Q+E. for Microsoft. Excel

User's Guide

Microsoft

Q+E™ for Microsoft® Excel

User's Guide

*Spreadsheet with Business Graphics and Database
Version 3.0*

OS/2 Series and Windows™ Series

Microsoft Corporation

Table of Contents

Before You Begin

Q+E™ is an easy-to-use yet powerful database application that allows you to manipulate and update database files from a variety of database systems. These include dBASE®, Microsoft® SQL Server, Oracle®, and OS/2 Extended Edition, as well as text files and Microsoft® Excel files containing a defined database.

Q+E makes working with your database files a breeze. By using Q+E's intuitive menu structure, the database novice, as well as the expert, can get right to work creating reports and updating records. And, since Q+E runs under Microsoft Windows or OS/2, you can take full advantage of the windows environment. You can open several database files at once, and copy data from one to another or to a different application. If you copy data to Microsoft Excel, or any other application that supports Dynamic Data Exchange (DDE), you can link the data to Q+E so that the data is updated automatically whenever the database information changes.

Querying & Editing

When you open a database file in Q+E, its records are displayed in a Query window. In this window, you can construct a question, or *query*, to get specific information from your database files. You can sort, select, and format records, as well as make calculations to analyze your data. All your command choices can be saved in a query file, so you can quickly execute them again to reproduce the same view with the most up-to-date information. When you want to create a report, you simply print the results displayed in the Query window.

Q+E also provides tools to edit and maintain your database files. You can update fields, and add and delete records using the custom view you have created. You can also create new database files from scratch, or modify or delete the field definitions in existing ones.

Using This Guide

The *Q+E for Microsoft Excel User's Guide* is organized to make Q+E easy to learn, as well as to provide a convenient reference guide.

- Part I, "Getting Started," explains how to install and start Q+E, and how to open database files and look at your data.

- Part II, "Querying Database Files," explains how to define queries to get specific information from your database files, and how to format, save, and print the results.

- Part III, "Maintaining Database Files," explains how to update, add, and delete records, and how to create new database files and modify or delete existing ones.

- Part IV, "Using Q+E with Other Applications," explains how to share database information with other applications via DDE. This includes linking other applications to Q+E, working with Q+E data from within Microsoft Excel, and writing macros to communicate with Q+E.

- Appendixes document any Q+E features that are unique to a particular database system Q+E supports. Wherever possible, Q+E features and commands are documented generically in the main part of this manual.

Sample Files

Several sample dBASE files and text files are included on the Q+E disk. The examples in this manual are based on these sample files, and you can open them at any time to try out procedures yourself. When you install Q+E, the sample files are copied to the same directory that contains the Q+E program files. If you want to open these files, make sure you install the dBASE driver and the Text File driver when you install Q+E.

Part 1

Getting Started

Q+E Basics

This chapter explains how to install and start Q+E, open and close database files, use online Help, browse and select data, and quit Q+E.

Installing Q+E

You can install Q+E at the same time you install Microsoft Excel. It is one of your options when you run the Microsoft Excel Setup program. If you chose not to install Q+E when you installed Microsoft Excel, run the Setup program again, select the Custom Installation option, and then select the Q+E option. Setup will prompt you for information and then install Q+E.

For information on running the Microsoft Excel Setup program, see Chapter 1, "Microsoft Excel Basics," in the *Microsoft Excel User's Guide*.

The Setup program will ask if you want to install a read-only version of Q+E. Answer yes only if you do not want users to be able to change, add, or delete records.

The Setup program will also ask you which database systems you plan to use with Q+E. For each database system you choose, the Setup program will copy a database driver onto your hard disk and modify your WIN.INI or OS2.INI file so that Q+E can use the driver. If you want to open the sample dBASE and text files provided with Q+E, be sure to install the dBASE and Text File drivers.

After you have installed Q+E, you can run the Setup program again at any time to install additional database drivers.

Starting Q+E

After you have installed Q+E, you can start the application.

NOTE   You can start Q+E directly from Microsoft Excel if you open the QE.XLA add-in macro. For more information, see Chapter 9, "Using Microsoft Excel to Access Data on External Databases."

ⅢⅢ Starting Q+E

1   If you are using Windows, open the Windows Applications group in the Program Manager window or open the group in which Q+E is stored.

If you are using OS/2, open the Applications group in the Desktop Manager window or open the group in which Q+E is stored.

2   Double-click the Q+E icon, or use the ARROW keys to select the Q+E icon and press ENTER.

Q+E starts and displays the main Q+E window. From this window, you can open database files.

The procedures for using commands and dialog boxes in Q+E are the same as those for any other Windows or OS/2 application. For information on choosing menu commands, see Chapter 1, "Microsoft Excel Basics," in the *Microsoft Excel User's Guide*.

Opening Database Files

To query or edit a database file, you open it in a Query window. For example, you could open the sample dBASE file EMP.DBF in a Query window to see all the records it contains. You can have several database files open at one time in different Query windows.

This section provides a general procedure for opening database files. The options will vary depending on the source database system. In addition, if you want to open SQL Server, Oracle, or Extended Edition tables, you must first log on to a database server. For details on opening a particular type of database file or logging on to a server, see the corresponding appendix.

For details on opening	See
dBASE database files	Appendix A, "Using Q+E with dBASE"
Text files	Appendix B, "Using Q+E with Text Files"
SQL Server tables	Appendix C, "Using Q+E with SQL Server"
Oracle tables	Appendix D, "Using Q+E with Oracle"
Extended Edition tables	Appendix E, "Using Q+E with OS/2 Extended Edition"
Microsoft Excel worksheet files	Appendix F, "Using Q+E with Microsoft Excel Worksheet Files"

NOTE   In the main part of this manual, database files, tables, text files, and worksheet files are referred to generically as database files.

ⅢⅢⅠ Opening a database file

1   Choose File Open.

2   If you are running Q+E under Windows, select the source you want to access in the Source box. If you are running Q+E under OS/2, select the source in the Source List box and then choose OK. Q+E lists the different database systems for which drivers have been installed and any computer servers that have been logged on to (for example, when using SQL Server).

3   In the Directory List box, select the directory that contains the file you want to open and choose OK. Depending on the source, this box may have a different name.

4   In the File List box, select the database file you want to open. This box lists all files for the current source that are in the current directory. Depending on the source, this box may have a different name.

5   To specify additional options, choose the Options button, select the options you want, and choose OK.

6   Choose OK.

Q+E opens the file and displays its records in a Query window. If there are more database records or fields than can fit in the Query window, a scroll bar appears along the right side or bottom of the window. For information on moving around records, see "Moving and Selecting in a Query Window" later in this chapter.

If you have more than one Query window open, you can switch between the windows by pressing CTRL+F6, clicking a window, or choosing a window name from the Window menu.

You can also choose the Window Arrange All command to rearrange the Query windows so that you can see them all at once without overlapping.

The Query Window Menus

The following table summarizes the commands on the Query window menus.

NOTE   Additional commands may appear on the menus, depending on which database drivers are installed and what type of database file you are working with. For information on using commands that are specific to a particular source, see the appendix for that source.

This menu	Contains commands to perform these tasks:
File	Open database files and query files; define database files; save and print files.
Edit	Edit, move, and copy data; add and delete records.
Sort	Sort records.
Select	Select records that meet certain conditions; join database files; edit SQL statements.
Search	Search for a character string; go to a specific record number.
Layout	Move, remove, and add columns; edit column definitions; change the display font and column widths; calculate column totals.
Window	Change the active Q+E window; rearrange the open Q+E windows so that you can see them all at once; display a summary of the current query.
Help	Get information on using Q+E commands.

Some Q+E commands have shortcut keys. Shortcut key sequences are shown to the right of the corresponding command names in the menus.

Using Online Help

Q+E's online Help describes each Q+E menu command in detail. You can display help information on your screen at any time by pressing FI or by choosing the Help Index command from the Help menu. Online Help for Q+E is set up the same way that Microsoft Excel Help is. For more information on using Help, choose the Help Using Help command or see "Using Online Help," in Chapter 1, "Microsoft Excel Basics," in the *Microsoft Excel User's Guide*.

Moving and Selecting in a Query Window

Before carrying out most commands and tasks in Q+E, you must first select the part of the database file you want to work with. You can move around the Query window and select characters, fields, and records using either the keyboard or the mouse. Q+E also provides commands to "zoom" a value in a separate window, search for a character string, or jump to a specific record number.

Using the Mouse

The mouse techniques for moving and selecting are the same as those for most Windows and OS/2 applications.

To	Do this:
Scroll through the Query window	Click the scroll bar arrows.
Select a field	Click the field.
Select a range of fields	Drag through the range.
Select multiple fields or ranges	Select the first range; then hold down CTRL and select each additional range.
Select characters within a field	Click the field; then drag to select the characters.

If you have a mouse with multiple buttons, Q+E uses only the leftmost button. For more information on using the mouse, see Chapter 1, "Microsoft Excel Basics," in the *Microsoft Excel User's Guide*.

Using the Keyboard

To move between fields and records:

Press	To move to
TAB	Next field
SHIFT+TAB	Previous field
HOME	First field

Press	To move to
END	Last field
CTRL+HOME	First record
CTRL+END	Last record
DOWN ARROW	Next record
UP ARROW	Previous record

To scroll the Query window if the database file is too large to display all at once:

Press	To scroll
PAGE UP	Up one screen
PAGE DOWN	Down one screen
CTRL+PAGE UP	Left one screen
CTRL+PAGE DOWN	Right one screen

To select characters, fields, or records:

Press	To select
SHIFT+LEFT ARROW	Character to the left
SHIFT+RIGHT ARROW	Character to the right
SHIFT+SPACEBAR	Entire record
CTRL+SPACEBAR	Entire column
SHIFT+CTRL+SPACEBAR	Entire window

///// Selecting a range of fields

1 Select the first field you want and press F8.

2 Press the ARROW keys to select the remaining fields.

3 Press F8 again.

///// Selecting multiple ranges

1 Move to the first field in a range you want to select and press F8.

2 Press the ARROW keys to include the fields you want.

3 If you want to add another range, press SHIFT+F8 to keep the selection and then repeat steps 1 through 3.

If you are done selecting ranges, press F8.

Zooming a Field

Sometimes a value may be wider than its column. Numbers that are too wide are displayed as "####." For dates and characters, you see only the leftmost part of the value. After you select a value that is wider than its column, you can use the LEFT and RIGHT ARROW keys to scroll through a value. Or, you can "zoom" the field and display the entire value in a separate window.

ⅢⅢ Zooming a field

▶ Double-click the field. If you then double-click a different value, Q+E immediately displays that value in the zoom window.

▶ Select the field and choose Edit Zoom Field.

ⅢⅢ Closing the zoom window

▶ Choose the Close command from the zoom window's Control menu.

Text fields may contain linefeed and carriage return characters, which control where the line breaks are in the zoom window. When you look at the same text in the Query window, black boxes appear in place of the special characters.

For information on changing the width of a column, see Chapter 6, "Editing Records."

Searching for a Character String

You can search the records in a Query window for a string of characters just as you can in a word-processing document.

ⅢⅢ Searching for a character string

1 Select a field in each column you want to search.

2 Choose Search Find.

3 In the Text box, type the characters you want to search for.

4 Choose OK.

Q+E selects the first occurrence.

5 Choose Search Find Next to find the next occurrence.

Choose Search Find Previous to find the previous occurrence.

Moving to a Specific Record Number

If you know the number of a record, you can move immediately to that record using the Search Goto command. Record numbers are displayed at the left edge of the window. When you use Q+E to sort or select records, Q+E renumbers the records to reflect the new display order.

IIIII   *Moving to a specific record number*

1 Choose Search Goto.

2 In the Record Number box, type the number of the record to which you want to go.

3 Choose OK.

Closing a Query Window

Before you close a Query window, make sure you save any query results you want to keep. You are not automatically prompted to save. For more information, see Chapter 4, "Saving a Query, Saving Results, and Printing."

IIIII   *Closing a Query window*

▷ Choose File Close.

Quitting Q+E

When you quit Q+E, the application automatically closes any Query windows that are open on your screen.

IIIII   *Quitting Q+E*

▷ Choose File Exit.

Part 2

Querying Database Files

Chapter 2 *Sorting and Selecting Records*

This chapter explains how to sort and select records in the Query window, display data from two or more database files in one window, and display a summary of the current query.

All your sorting and selecting instructions can be saved in a query file, or the results can be printed to create a report. For more information, see Chapter 4, "Saving a Query, Saving Results, and Printing."

Sorting Records

When you open a database file in a Query window, Q+E displays the records in the order that they are stored in the database file. You can display the records in a different order by sorting on one or more columns of the database file.

After you specify a sort order, you must remove the sorting before you can specify a different sort order.

Sorting records by one column

1   In the Query window, select any field in the column on which you want to sort. For example, to sort EMP.DBF by last name, you would select any field in the LAST_NAME column.

2   To sort the records in ascending order, choose Sort Ascending. To sort the records in descending order, choose Sort Descending.

To remove the sorting, choose the Sort Reset Sort command.

Sorting records by multiple columns

You may want to sort using more than one column. For example, to display employee records first grouped by department and then sorted by salary within each department, you would first sort on the DEPT column and then sort on the SALARY column.

1   In the Query window, select any field in the first column by which you want to sort.

2   To sort the records in ascending order, choose Sort Ascending. To sort the records in descending order, choose Sort Descending.

3   Repeat steps 1 and 2 to sort the records within each previous grouping.

Another way to sort by multiple columns is to first select a field in each column and then choose the Sort Ascending or Sort Descending command once. For example, you could click the DEPT column, hold down CTRL and click the SALARY column, and then choose Sort Ascending. Q+E sorts by the first field you selected, then the second, and so on.

You can view the current sort order by choosing the Window Show Info command. For more information, see "Displaying a Summary of the Current Query" later in this chapter.

ⅢⅢ   Removing sorting

You must first remove any existing sort conditions whenever you want to define a new sort order. Q+E redisplays the records in their original order.

▷   Choose Sort Reset Sort.

Selecting Records

You can use the Select Add Condition command to select and display a subset of the records in a Query window. When you use this command, you specify a condition that the records must meet. For example, the condition could be that the DEPT column must contain the value D202. Or, the condition could be that the SALARY column must contain a value greater than $30,000. You can select records based on one condition or as many conditions as you want. Only records that satisfy the specified conditions are displayed in the Query window.

ⅢⅢ   Selecting records based on one condition

1   In the Query window, select a field in the column you want to constrain. If possible, select the value that you want to be part of the selection condition. For example, to display records that contain the value D101 in the DEPT column, you would select the value D101 in the DEPT column.

2   Choose Select Add Condition to display the Add Condition dialog box.

```
┌──────────────────────────────────────────────┐
│ Add Condition                    ┌──────────┐  │
│                                  │    OK    │  │
│ Column:  DEPT                    └──────────┘  │
│                                  ┌──────────┐  │
│ ┌─Operator─────────────────────  │  Cancel  │  │
│ │                                └──────────┘  │
│ │  ◉ Equal         ○ Not Equal               │
│ │                                             │
│ │  ○ Less Than     ○ Less or Equal           │
│ │                                             │
│ │  ○ Greater Than  ○ Greater or Equal        │
│ │                                             │
│ │  ○ Like          ○ Not Like                │
│ │                                             │
│ ├─────────────────────────────────────────── │
│  Value:  │ D101                             │ │
│                                                │
│ ☒ Case Sensitive                              │
└──────────────────────────────────────────────┘
```

In this dialog box, you specify a condition that the records must meet to be displayed. For example, in the Add Condition dialog box shown above, the DEPT column must contain ("Equal") the value D101 for a record to be selected.

3   Under Operator, select the comparison operator that you want to use to define the condition. The default operator is Equal. For information on LIKE and NOT LIKE, see "Selecting Records Using Pattern Matching" later in this section.

4   In the Value box, enter the value that you want used in the condition. By default, Q+E displays the value in the selected field.

5   Turn off the Case Sensitive box if you do not want Q+E to treat uppercase and lowercase letters as different letters.

6   Choose OK.

Q+E selects and displays only those records that meet the condition specified in the Add Condition dialog box.

To remove the condition and redisplay all the records in the database file, choose the Select Reset Conditions command.

▥ *Selecting records based on multiple conditions*

To build a selection with more than one condition, repeat the Select Add Condition command for each additional condition. For example, if you want to select employees who work in department D202 and make a salary greater than $30,000, you would choose Select Add Condition twice—once for each condition. You can join as many conditions as you want.

If you choose the Edit Undo command immediately after you add a condition, Q+E will remove that condition but preserve any previous conditions.

1   In the Query window, select a field in the first column you want to constrain. If possible, select a field that contains the value that you want to be part of the selection condition.

2   Choose Select Add Condition, fill in the Add Condition dialog box, and choose OK. Q+E displays the subset of records that meet the first condition.

3   Select a field in the column that you want to be part of the next selection condition.

4   Choose Select Add Condition. Note that the Add Condition dialog box now has a Connection option group, with buttons labeled AND and OR. These buttons appear on second and subsequent uses of the Select Add Condition command.

5 Select the AND button if you want to display a record only if it meets all previous conditions and the new condition you are adding. Select the OR button if you want to display records that meet all previous conditions or meet the new condition you are adding.

6 Fill in the rest of the Add Condition dialog box, and then choose OK. Q+E displays the subset of records that meet the combination of selection conditions you have specified.

7 Repeat steps 3 through 6 to add as many selection conditions as you want.

If you add more than two conditions, Q+E evaluates the conditions in the order that you add them. For example:

These selection conditions	Return
condition OR *condition*	Records that meet either condition.
condition OR *condition* AND *condition*	Records that meet either of the first two conditions and also meet the third condition.
condition AND *condition* OR *condition*	Records that either meet both of the first two conditions or meet the third condition.

You can view the current selection conditions by choosing the Window Show Info command. For more information, see "Displaying a Summary of the Current Query" later in this chapter.

ⅢⅢ *Removing conditions*

You can remove selection conditions to redisplay all the records in a database file. You must remove any existing conditions whenever you want to start a new selection.

▶ Choose Select Reset Conditions.

Selecting Records Using Pattern Matching

Most of the comparison operators in the Add Condition dialog box are self-explanatory. Two that require special mention are LIKE and NOT LIKE.

LIKE allows you to select records using the standard pattern-matching (wildcard) characters.

If you are using MS-DOS pattern matching:

This character	Matches
*	Zero or more characters.
?	Any single character.

If you are using SQL pattern matching:

This character	Matches
%	Zero or more characters.
_	Any single character.

To specify whether you want to use MS-DOS or SQL pattern matching, choose the Layout Options command and select the type you want in the Options dialog box.

The following table shows how you could use MS-DOS pattern matching to search a LAST_NAME column for specific records. The value on the left is entered in the Value box.

Value	Matches
B*	Names that start with the letter B.
*N	Names that end with the letter N.
B*N*T	Names that start with B, contain an N, and end with a T.
?????	Names that have five characters.
*A?	Names whose next to last character is an A.

NOT LIKE is the opposite of LIKE. It is used to select all records that do not match the specified pattern.

Selecting Data from Multiple Database Files

Q+E lets you display data from multiple database files in one Query window using the Join command. For example, if your company has a department database and an employee database, you may want to join these files to display the employee records grouped by the department.

	DEPT_ID	DEPT_NAME	LOC_ID	MGR_ID	FIRST_NAME	LAST_NAME	EMP_ID	HIRE_
1	D190	Accounting	L31	E10001	Kim	Arlich	E10001	07/30,
2					Timothy	Grove	E16398	01/21,
3	D202	Marketing	L04	E01234	Adam	Smith	E63535	01/15,
4					Rich	Holcomb	E01234	06/01,
5					David	Motsinger	E27002	05/05,
6	D050	Sales	L31	E21437	John	Rappl	E21437	07/15,
7					Nathan	Adams	E41298	02/15,
8	D101	R&D	L23	E00127	Tyler	Bennett	E10297	06/01,
9					George	Woltman	E00127	08/07,
10					David	McClellan	E04242	07/27,
11					Richard	Potter	E43128	04/12,
12					Tim	Sampair	E03033	12/02,

Q+E — File Edit Sort Select Search Layout Window — Help
Query1 [DEPT.DBF, EMP.DBF]

To join two database files in a single Query window, they must have columns in common. For example, DEPT.DBF and EMP.DBF could be joined (as shown above) because they each have a Department column with common values. The columns do not have to have the same name (they are DEPT_ID and DEPT in this example), but they must contain the same values so that Q+E can match the records.

When you join two files, you join the data in one window (the source) to the data in another window (the destination). The destination records are related to the source records in one of two ways:

One-to-One Relationship   In this type of relationship, each destination record will match one source record. For example, for each department record, there is one employee who is the manager of the department.

One-to-Many Relationship   In a one-to-many relationship, each destination record may match any number of source records. For example, there are a number of employees working in each department. The join shown above is a one-to-many relationship.

You can join additional windows to the results of a join, combining three or more files.

Joining Database Files

You can join files using either the Select Join command or the Select Outer Join command. Either command will join the source window to the destination window and display the results. However, Select Join discards all records in the destination window that have no matching records in the source window, whereas Select Outer Join shows all records from the destination window whether they match or not.

The following procedure outlines the general steps for joining two database files. The procedure is the same for one-to-one relationships or one-to-many relationships. The type of relationship is determined by the contents of the database files. For specific examples of different types of joins, see "Examples of Joins" below.

▐▐▐▐▐   Joining database files

1   Use File Open to open the database files you want to join.

2   Optionally, choose Window Arrange All so that you can view both files at once.

3   In the source window, select the column that contains values in common with the destination window.

4   In the destination window, select the column to match. The destination window must be the active window.

5   Choose Select Join to join the source window to the destination window and display the results. Select Join discards all records in the destination window that have no matching records in the source window.

Choose Select Outer Join if you want to show all records from the destination window whether they match or not.

Examples of Joins

This section provides examples of joining files using the sample dBASE files that are included with Q+E. If you are unfamiliar with relational join operations, it is recommended that you work through these exercises to learn how to add joins to your queries.

ⅠⅠⅠⅠ   Creating a one-to-one relationship

In this example, you join LOC.DBF (the source) to DEPT.DBF (the destination) so that you can list the location of each department and other data in one Query window.

1   Open the file named DEPT.DBF (the department records). Note that the record for Production does not include a value in the LOC_ID column.

2   Open the file named LOC.DBF (the location records).

3   Choose Window Arrange All. Note that both files contain a common LOC_ID column but only LOC.DBF contains the city and state of the location.

4   Select any value in the LOC_ID column of LOC.DBF (the source).

5   Select any value in the LOC_ID column of DEPT.DBF (the destination).

6   Choose Select Join. The location records are joined to the department records based on common values in the LOC_ID columns.

In one Query window, Q+E displays all the data from DEPT.DBF and the city and state information from LOC.DBF.

Note that Production does not appear in the result. This is because the record for Production does not have a LOC_ID value that matches a value in the location file or, in this case, any location value at all. If you want to see all department records regardless of whether they have a matching record in the location file, follow steps 1 through 5 above and then choose the Select Outer Join command instead of the Select Join command.

ⅠⅠⅠⅠ   Creating a one-to-many relationship

In this example, you join EMP.DBF (the source) to DEPT.DBF (the destination) so that you can list each department with all its employees in one Query window.

1   Open the file named DEPT.DBF (the department records).

2   Open the file named EMP.DBF (the employee records).

3   Choose Window Arrange All. Note that both files contain a department iden-
     tifier column: DEPT_ID for the department file and DEPT for the employee
     file.

4   Select any value in the DEPT column of EMP.DBF (the source).

5   Select any value in the DEPT_ID column of DEPT.DBF (the destination).

6   Choose Select Join. The employee records are joined to the department
     records based on common values in the DEPT and DEPT_ID columns.

In one Query window, Q+E displays the department records from DEPT.DBF
and several employee records from EMP.DBF for each department record (see
the illustration at the beginning of this section). This is a one-to-many relation-
ship because for each department record, there may be many employee records.
Note that duplicate values are displayed only once.

Production contains no employees, so its department record is not displayed in
the final result. If you want to include records for departments that have no
employees, use the Select Outer Join command instead of the Select Join
command.

ⅢⅢ   Creating multiple joins

In this example, you first join LOC.DBF to DEPT.DBF (a one-to-one relation-
ship), then join EMP.DBF to the results (a one-to-many relationship). This lets
you view the departments, their locations, and their employees in a single Query
window.

1   Join LOC.DBF to DEPT.DBF on their shared location values using the Select
     Join command (follow the steps under "One-to-One Relationship" above).
     Note that the title bar in the Query window now reads DEPT.DBF,LOC.DBF.

2   Open EMP.DBF, and then select any value in the DEPT column (the source).

3   Make DEPT.DBF,LOC.DBF the active Query window, and then select any
     value in the DEPT_ID column (the destination).

4   Choose Select Join.

In one Query window, Q+E displays the department records from DEPT.DBF,
the city and state information from LOC.DBF, and the employee records from
EMP.DBF for each department record.

When you join three or more files, the order in which you join the files may affect
the results. For example, if you joined EMP.DBF to DEPT.DBF and then joined
LOC.DBF to the results, the city and state information would be repeated for each
employee, not just for each department as in the example above.

In general, you should first construct all joins that form one-to-one relationships
and then add the one-to-many relationships. This sequence avoids repeating in-
formation in one-to-one relationships.

Displaying a Summary of the Current Query

You can use the Window Show Info command to display summary information about the current query in a separate window. For example, if you sorted EMP.DBF and then selected employees whose department is D202 or who have a salary greater than $40,000, the Show Info window would look something like this:
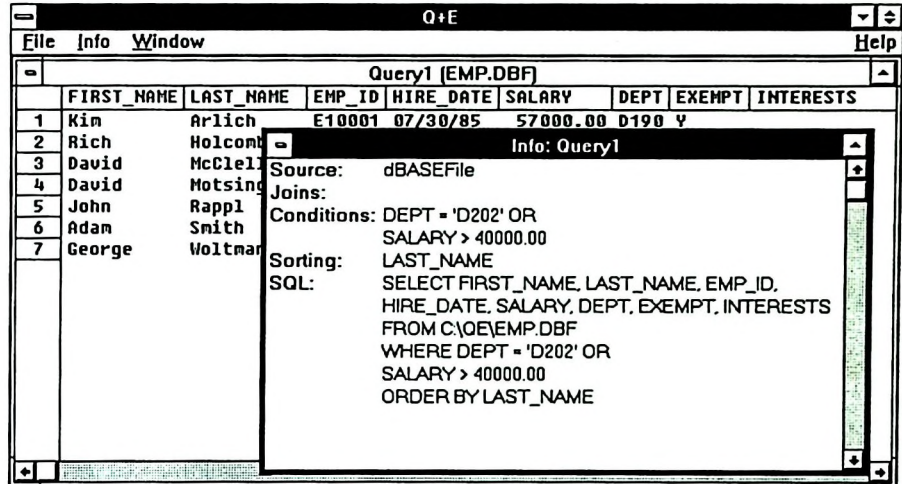
The Show Info window shows the current database system, joins, selection conditions, and sort order, as well as the current SQL SELECT statement. SQL is a standard database language that is used by a number of database programs. As you use the commands in the Query window to manipulate database records, Q+E automatically creates and maintains the corresponding SQL SELECT statement. For more information on SQL, see Chapter 5, "Editing the Current SQL Statement."

In the Conditions section, comparison operators are expressed by the following symbols:

Symbol	Operator
=	Equal
<> or !=	Not equal
>	Greater than
>=	Greater than or equal
<	Less than
<=	Less than or equal
LIKE	Like
NOT LIKE	Not like

ııııı *Opening the Show Info window*

1 Activate the Query window for which you want summary information.

2 Choose Window Show Info.

When the Show Info window is the active window, the Show Info menus replace the Query window menus. You can choose commands from the Info menu to specify what information you want displayed in the Show Info window.

ııııı *Closing the Show Info window*

1 Activate the Show Info window.

2 Choose File Close from the Show Info window's Control menu.

Chapter 3   *Formatting and Analyzing Your Data*

After you sort and select records in a Query window, you can format the data and make calculations to analyze your data. This chapter first explains how to format columns and rows in the Query window, such as changing column widths, moving columns, and so on. It then explains how to add a computed column and how to calculate column totals.

All your formatting and calculations can be saved in a query file or printed to create a report. For more information, see Chapter 4, "Saving a Query, Saving Results, and Printing."

Formatting the Display

In a Query window, you can change column widths, change column headings, move and hide columns, hide duplicate rows, and change the character font. All these changes can be saved in a query file but do not affect the database file itself.

Changing Column Widths

When you open a file in a Query window, Q+E sets each column to its default column width. This is either the width defined for the field in the database file or the maximum default column width set for that field type, whichever is smaller.

In the Query window, you can override the default column width and make each column as wide or as narrow as you want. You can also change the maximum default column widths defined for the character, number, and date field types. For information on changing field widths in a database file, see Chapter 7, "Defining Database Files."

ⅢⅢ   *Changing the width of a column*

▶ Point to the vertical bar to the right of the column heading, and drag to the desired column width.

1   Select any field in the column, and then choose Layout Column Width.

2   In the Column Width box, type the number of characters or numeric digits you want to fit in the column. The actual number of characters displayed in character fields may vary, since characters in proportional fonts vary in size.

3   If you want to set the column width to the default size, turn on the Default Width check box. The default size is the width of the field in the database file or the maximum default column width, whichever is smaller.

4   Choose OK.

///// *Changing the maximum default column widths*

1 Choose Layout Options.

2 Under Maximum Default Width, enter the maximum default column width for character, number, and date fields.

3 Choose OK.

Changing Column Headings

In a Query window, Q+E uses the field names from the database file as the default column headings. You can change the column headings to customize the display.

///// *Changing a column heading*

1 Double-click the column heading, or select any field in the column and choose Layout Define Column.

2 In the Heading box, type the column heading you want.

3 Choose OK.

You can also use the Layout Define Column command to add a new column whose values are derived from data in other columns. For more information, see "Creating a Computed Column" later in this chapter.

Moving Columns

When you open a database file, Q+E displays columns in the order that the fields appear in the database file. In the Query window, you can change the order of the columns in any way you want.

///// *Moving a column*

1 Select any field in the column you want to move.

2 Choose Layout Move Column. The cursor changes shape.

3   Point to where you want to insert the column and click.

  Press TAB or SHIFT+TAB to move to a column adjacent to where you want to insert the column. Then, press the LEFT ARROW key or the RIGHT ARROW key to place the column to the left or right of this column.

If you want to reorder two or more columns, select each column in the order you want it to appear in the Query window and then choose the Layout Move Column command.

Removing and Restoring Columns

When you open a database file, Q+E displays a column for every field in the database file. You can remove columns in the Query window to display only the data you need. After you remove a column, you can restore it using the Layout Define Column command.

ⅠⅠⅠⅠ Removing a column

1   Select any field in the column. To remove more than one column, select a field in each column.

2   Choose Layout Remove Column.

ⅠⅠⅠⅠ Restoring a column

1   Choose Layout Define Column.

2   In the Fields box, select the column you want to restore. The Expression box shows the database field name.

3   If you want the column heading to be different than the database field name, type the heading in the Heading box. If you do not enter a name in the Heading box, the column heading will be the database field name by default.

4   Choose the Add button. The cursor changes shape.

5   ✸ Point to where you want to insert the restored column and click.

    ✏ Press TAB or SHIFT+TAB to move to a column adjacent to where you want to insert the column. Then, press the LEFT ARROW key or the RIGHT ARROW key to place the column to the left or right of this column.

You can also add a new column whose values are derived from data in other columns. For more information, see "Creating a Computed Column" later in this chapter.

Hiding and Unhiding Duplicate Rows

If you are formatting data from SQL Server, Oracle, or OS/2 Extended Edition, you can hide duplicate rows in the Query window—if two or more rows of data are identical, Q+E will display only one of the rows.

ⅠⅠⅠⅠ Hiding or unhiding duplicate rows

▶   Choose Select Distinct.

When duplicate rows are hidden, a check mark appears next to the command on the menu. Choose Select Distinct again to redisplay duplicate records.

Changing the Character Font

You can change the format of text in a Query window. Depending on your printer setup, you can choose from a variety of fonts and point sizes, and format text as bold, italic, underlined, or strikeout.

ⅢⅢ   Changing the character font

1   Choose Layout Font.

2   Select the font and point size you want. You can enter a point size that is not an option in the list, but the Query window may not be redisplayed as quickly.

3   Under Style, turn on the check boxes for the formatting options you want.

4   If you want your settings to be the defaults for any text in new Query windows you open and to be the default the next time you run Q+E, turn on the Set Default check box.

5   Choose OK.

Setting Date, Time, Currency, and Number Formats

Q+E uses the International or Country settings in the Microsoft Windows or OS/2 Control Panel to control the formatting of dates, times, currency, and numbers. For more information on the Control Panel, see your Microsoft Windows or OS/2 documentation.

From within Q+E, you can specify whether to include a thousands separator in numbers, such as the comma in 10,000.

ⅢⅢ   Turning on or off the thousands separator

1   Choose Layout Options.

2   Turn on the Use Thousands Separator check box to include the thousands separator in numbers. Turn off the check box if you don't want the separator.

Creating a Computed Column

You can add a computed column to a Query window to get new information from your data. To create a computed column, you define an *expression*—a text or arithmetic formula—from which Q+E calculates values for the column. For example, suppose you want to know how much each employee contributes to a savings plan. Assuming that each employee contributes 8% of his or her salary to the savings plan, you could add a new column named SAVINGS CONTRIB and assign it this expression:

SALARY * .08

Q+E passes expressions to the underlying database system to evaluate. Therefore, you can write expressions using whatever operators and functions are supported by your database system. All database systems support the following mathematical operators:

Operator	Purpose
+	Add
-	Subtract
*	Multiply
/	Divide

For information on other operators and functions you can use, see the appendix for your database system or the system documentation.

///// **Adding a computed column**

1   Choose Layout Define Column.

2   In the Heading box, type the column heading.

3   In the Expression box, type the expression that defines the column's values.

4   Choose the Add button. The cursor changes shape.

5   Point to where you want to insert the computed column and click.

Press TAB or SHIFT+TAB to move to a column adjacent to where you want to insert the computed column. Then, press the LEFT or RIGHT ARROW key to place the column to the left or right of this column.

Calculating Column Totals

Q+E can perform calculations on selected columns and display totals in the Query window. For example, you could show sum, minimum, maximum, count, and average totals for the SALARY column in EMP.DBF.

	FIRST_NAME	LAST_NAME	EMP_ID	HIRE_DATE	SALARY	DEPT	EXEMPT	INTERESTS
1	Nathan	Adams	E41298	02/15/88	21900.00	D050	N	
2	Kim	Arlich	E10001	07/30/85	57000.00	D190	Y	
3	Tyler	Bennett	E10297	06/01/77	32000.00	D101	Y	
4	Timothy	Grove	E16398	01/21/85	29900.00	D190	Y	
5	Rich	Holcomb	E01234	06/01/83	49500.00	D202	Y	
6	David	McClellan	E04242	07/27/82	41500.00	D101	Y	Sports:■■
7	David	Motsinger	E27002	05/05/85	19250.00	D202	N	
8	Richard	Potter	E43128	04/12/86	15900.00	D101	N	
9	John	Rappl	E21437	07/15/87	47000.00	D050	Y	Sports:■■
10	Tim	Sampair	E03033	12/02/87	27000.00	D101	Y	
11	Adam	Smith	E63535	01/15/88	18000.00	D202	N	
12	George	Woltman	E00127	08/07/82	53500.00	D101	Y	Sports:■■
SUM					412450.00			
MIN					15900.00			
MAX					57000.00			
CNT					12			
AVG					34370.83			

You can show minimum, maximum, and count totals for character and date columns as well. For example, in the LAST_NAME column, the smallest value is the name closest to the beginning of the alphabet and the largest value is the name closest to the end of the alphabet.

ııııı   Computing totals for a column

1   Select any value in the column. If you select more than one column, Q+E will calculate the totals you specify for each column.

2   Choose Layout Totals.

3   Turn on the check boxes for the totals you want.

4   Choose OK.

Q+E displays column totals at the bottom of the database file. If necessary, scroll the Query window to see column totals.

If you calculate column totals and then change your selection conditions, Q+E recalculates the column totals to reflect your new selection.

ııııı   Removing totals for a column

1   Select any field in the column and choose Layout Totals.

2   Turn off the check boxes for the totals you do not want to display.

Displaying Totals Only

After you add column totals, you can hide the individual records used to compute the totals.

ııııı   Hiding or redisplaying the record detail

▶   Choose Layout Only Show Totals.

When the record detail is hidden, a check mark appears next to the command on the menu. Choose Layout Only Show Totals again to redisplay the records with the totals.

Chapter 4   *Saving a Query, Saving Results, and Printing*

This chapter explains how to save your work and print from Q+E. After you construct a query using the commands in the Query window, you can:

- Save the query definition so that you can execute the same query again as the data in your database files changes.

- Save the query results to a new database file.

- Save the results, formatted as mailing labels, to an ASCII text file.

- Print the results as they appear on screen, or save the results to a formatted ASCII text file.

Saving a Query

When you save a query in a query file, you are not saving the records that are displayed in the Query window. Instead, you are saving the names of the database files you opened and the Q+E commands you chose in order to get the set of records displayed in the window.

When you open a query file, Q+E executes the saved commands on the current information in the database files and displays the results in a Query window. By saving your queries in query files, you can quickly reproduce the same views and reports using the most up-to-date information.

Saving a query definition in a query file

When you save a query file, every action you performed to produce the current set of records in their current format is saved to the query file, including all calculations and display formatting. If you defined a label format, that is saved also. For more information on labels, see "Saving Results as Mailing Labels" later in this chapter.

1   In the Query window, choose File Save As.

2   If you are running Q+E under Windows, select QueryFile in the Destination box. If you are running Q+E under OS/2, select QueryFile in the Destination List box. The default name for the query file is the Query window name with the extension .QEF.

3   To specify a different name for the query file, type the name in the File Name box. Q+E will add the extension .QEF for you.

4   Choose OK.

NOTE   Q+E saves the SQL SELECT statement that defines the query in a query file. For more information, see Chapter 5, "Editing the Current SQL Statement."

IIIII *Opening a query file*

When you open a query file, Q+E repeats all the actions you saved in the query file and displays the results in a Query window. The title bar in the window displays the name of the query file.

1 Choose File Open.

2 Select the Query File.

3 Select the query file name in the File List box, or type the name in the File Name box.

4 Choose OK.

If you change the query in the Query window, you can save your changes using the File Save command.

Saving Results to a New Database File

To save the actual data in the Query window, you save the results to a new database file, which will contain the records and fields in the order they appear in the Query window. Computed columns are saved as regular columns in the new database file. For example, if the expression for a column named SALES_CONTRIB was SALARY*.08, the expression would become "SALARY_08" in a new database file (the latter involves no calculations).

NOTE Column totals are not saved to a new database file. To save this information, you must save to a query file or print to a text file.

You can save query results to any database system that Q+E supports. Q+E automatically converts the new database file to the appropriate database format.

IIIII *Saving query results to a new database file*

1 In the Query window, choose File Save As.

2 Select the database format in which you want to save the results. Q+E lists the different database systems for which drivers have been installed and, if necessary, have been logged on to.

3 In the File Name box, type a name for the new database file. Q+E will automatically add an extension when appropriate. Depending on the destination, this box may have a different name.

4 If you want the field names in the new database file to be the same as the column headings in the Query window, turn on the Use Headings for Field Names check box. Otherwise, the field names in the new database file will be the underlying expressions that define the columns (for example, the expression SALARY_08 instead of the column heading SALES_CONTRIB).

5   If options are available, choose the Options button, select the options you want, and choose OK. Options depend on the type of database file. For more information, see the appendix for your database system.

6   Choose OK.

Saving Results as Mailing Labels

You can use the File Save As command to format and save the results in the Query window as mailing labels. For example, ADDR.DBF contains employee names and addresses. You could save the data in ADDR.DBF in a mailing label format in a text file and then print the labels when you need them.

Creating a Label File

A label can be up to 128 lines long and contain any number of fields on each line. To create a label, you first select in the Query window the fields you want the label to include and then choose the File Save As command. Q+E sets up a default label based on your selections, which you can customize in the Label Definition dialog box.

▥▥▥ Creating a label file

1   Open the database file that contains the names and addresses you want to use for the labels.

2   If you want to include two or more fields on one line (for example, FIRST_NAME and LAST_NAME), make sure that the fields are next to each other in the Query window and in the order you want them on the mailing label. To add more than one field to a line, you must be able to enclose the fields in a single selection. If necessary, use the Layout Move Column command to reorder the columns in the Query window.

3   To specify the fields you want on the first line of the label, make a single selection that includes a field from each column. For example, using the mouse, you would drag over the FIRST_NAME and LAST_NAME fields.

4   To define the next line of the mailing label, keep your first selection and select the fields you want on the next line. To do this with the mouse, hold down the CTRL key and drag over the fields you want on the next line. For more information on making multiple selections, see Chapter 1, "Q+E Basics."

5   Repeat step 4 to specify each line of the label.

6   Choose File Save As.

7   Select the destination MailingLabels.

8   To modify the default label definition, choose the Options button, make your changes in the Label Definition dialog box, and choose OK. For more information, see "Modifying the Default Label Definition" below.

9 In the Save As box, type a name for the label file. Q+E will automatically add the extension .LAB.

10 Choose OK.

To view or print the mailing labels, open the .LAB file in a text editor, such as Windows Notepad.

Modifying the Default Label Definition

When you save a label file, you can modify the default label definition by choosing the Options button in the File Save As dialog box. Q+E displays the Label Definition dialog box.

Each line of the label definition corresponds to a line on the label. If there is more than one field on a line, Q+E inserts a blank space (' ') between each pair of fields and concatenates the field names and blank spaces with plus (+) signs.

You can customize the label definition by adding or deleting field names or string constants in the Label Definition dialog box. A string constant is a value that you want to appear on every label and must be surrounded by single quotation marks ('). For example, you could add a comma within the quotation marks between the CITY and STATE fields in the example above. Field names and string constants on one line must be separated by plus signs.

In the Lines Between box, you can indicate the number of blank lines you want after each label. The default is one.

Under Column Start Positions, you can indicate that you want to print up to four mailing labels across the page instead of printing one long column of labels. In the 1 box, type the column position where you want to start printing the first

column; in the 2 box, type the column position where you want to start printing the next column; and so on.

Under File Character Set, select the character set you want Q+E to use when it saves the labels in a .LAB file. The two character sets are about the same, although ANSI has better support for international characters.

Changing the Mailing Label Definition in a Query File

When you save a query file, the current label definition is saved also. You can easily replace the existing label definition with a new one.

▪▪▪▪▪ Changing the existing label definition in a query file

1 Open the query file.

2 In the Query window, select columns to define the new label definition.

3 Choose File Save As.

4 Select the destination MailingLabels, and then choose the Options button.

5 Choose the New button.

6 Make any other changes you want in the Label Definition dialog box.

7 Save the query file.

Printing Results

You can print query results at any time. The records are printed in formatted columns as they appear in the Query window. If there are more columns than fit the width of your printer, Q+E prints the extra columns on following pages.

In addition to printing hard copy, you can save query results to a formatted text file.

▪▪▪▪▪ Setting up your printer

1 Choose File Printer Setup.

2 Select the printer you want.

3 To change the printer setup, choose the Setup button, specify the settings you want, and choose OK. See your printer's documentation for more information.

4 Choose OK.

▪▪▪▪▪ Setting up the page format

1 Choose File Page Setup.

2 In the Title box, type the page header you want centered at the top of each printed page.

3   Under Margins, specify the margin settings you want. The margins are given in inches.

4   Under Include Options, turn on the check boxes for the items you want to appear on each printed page.

ⅡⅡⅡⅡ   *Printing the results*

1   Choose File Print.

2   In the Copies box, type the number of copies you want to print.

3   If you want to print a draft copy quickly, instead of a final copy, turn on the Draft Quality box.

4   Choose OK.

ⅡⅡⅡⅡ   *Printing to a file*

When you direct printer output to a text file, Q+E saves the results using the Title and Include Options settings in the File Page Setup dialog box.

1   Choose File Save As.

2   Select the destination PrintToFile.

3   To set the page width and lines per page, choose the Options button, enter the values you want, and then choose OK.

4   Choose OK.

Chapter 5   *Editing the Current SQL Statement*

As you use the commands in the Query window to manipulate your records, Q+E maintains the corresponding SQL SELECT statement. SQL (Structured Query Language) is a standard database language used to build queries in many database systems. With Q+E, you can construct a SQL query by choosing menu commands or you can edit or write the SQL statement directly in a SQL Query window.

This chapter explains how to view and edit the current SQL statement. It also explains how to define a query from scratch using SQL.
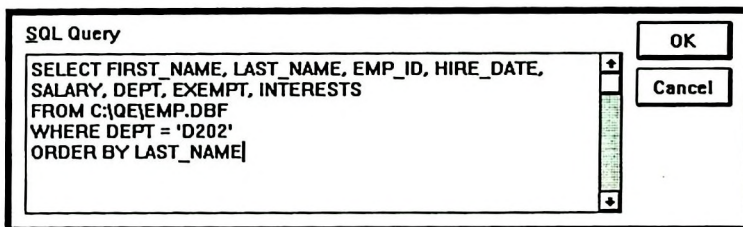
About the SQL SELECT Statement

You can view the current SQL SELECT statement for a Query window at any time by choosing the Select SQL Query command.

At a minimum, the SQL statement includes two clauses, SELECT and FROM. SELECT specifies which columns you want to display, and FROM specifies which database files contain the columns. For example, the following SQL Query window shows the SQL SELECT statement for EMP.DBF with all its fields displayed.

```
 SQL Query                                              ┌──────────┐
┌─────────────────────────────────────────────┬─┬─┐    │    OK    │
│ SELECT FIRST_NAME, LAST_NAME, EMP_ID, HIRE_DATE, │▲│    └──────────┘
│ SALARY, DEPT, EXEMPT, INTERESTS                  │ │    ┌──────────┐
│ FROM C:\QE\EMP.DBF│                              │ │    │  Cancel  │
│                                                  │ │    └──────────┘
│                                                  │ │
│                                                  │▼│
└─────────────────────────────────────────────┴─┴─┘
```

When you define a sort order, add selection conditions, and so on, other clauses are added. For example, when you sort records using the Sort menu, Q+E adds an ORDER BY clause, and when you add selection conditions using the Select menu, Q+E adds a WHERE clause.

```
 SQL Query                                              ┌──────────┐
┌─────────────────────────────────────────────┬─┬─┐    │    OK    │
│ SELECT FIRST_NAME, LAST_NAME, EMP_ID, HIRE_DATE, │▲│    └──────────┘
│ SALARY, DEPT, EXEMPT, INTERESTS                  │ │    ┌──────────┐
│ FROM C:\QE\EMP.DBF                               │ │    │  Cancel  │
│ WHERE DEPT = 'D202'                              │ │    └──────────┘
│ ORDER BY LAST_NAME│                              │ │
│                                                  │▼│
└─────────────────────────────────────────────┴─┴─┘
```

When you choose menu commands, Q+E builds the SELECT statement using its own version of SQL, although you can write a SQL statement yourself using any SQL that is supported by the underlying database system. The following table summarizes which menu commands in the Query window add to or change the current Q+E SQL SELECT statement.

This part of a Q+E SQL statement	Is added or changed by
SELECT statement	File Open, Select Add Condition, Select Reset Conditions, Select Join, Select Outer Join, Layout Define Column, Layout Move Column, Layout Remove Column
DISTINCT keyword	Select Distinct
FROM clause	File Open, Select Join, Select Outer Join, File Use Index
WHERE clause	Select Add Condition, Select Reset Conditions
ORDER BY clause	Sort Ascending, Sort Descending, Sort Reset Sort
COMPUTE clause	Layout Totals
OPTIONS clause	Layout Only Show Totals

SQL Expressions

If you add a computed column to the Query window, the expression that defines the computed column appears in the SQL statement. For example, when you create a computed column using the Layout Define Column command, you enter an expression (such as "SALARY * .08") to define the values in the column. This expression becomes a part of the SQL SELECT statement.

SQL expressions are also used to define various conditions in the other clauses of SQL statements. For example, if you used the Layout Totals command to compute an average salary, it would be defined by the expression AVG(SALARY) in the COMPUTE clause of the SQL statement.

Editing the Current SQL Statement

Most of the time it is easiest to choose commands in the Query window to define a query. In some cases, however, you may find it easier to edit the SQL statement itself. Or, you may want to edit the SQL statement to define an advanced query that is not directly supported by the Q+E interface.

For example, the selection condition in the following SQL SELECT statement can be defined only by editing the SQL statement directly:

```
SELECT FIRST_NAME,LAST_NAME FROM EMP
WHERE DATE() – HIRE_DATE >= 90
```

This statement retrieves records for employees who have been with the company at least 90 days. You could use the menu commands to select the FIRST_NAME and LAST_NAME fields from EMP.DBF, but you would need to edit the statement to specify the selection condition. The DATE() function can be used since it is supported by dBASE.

Here is another example:

```
SELECT FIRST_NAME,LAST_NAME FROM EMP
WHERE MONTH(HIRE_DATE) = 1 AND (YEAR(HIRE_DATE) = YEAR(DATE())
OR YEAR(HIRE_DATE) = YEAR(DATE()) —1)
```

This statement retrieves from EMP.DBF all employees that were hired in January of this year or in January of last year.

ⅢⅢ Editing the current SQL SELECT statement

When you edit a SQL statement, you can add any SQL clauses and expressions that are supported by the Q+E driver for the source database system. For more information, see the appendix for the database system.

1 Choose Select SQL Query or double-click the box to the left of the first column heading.

2 Make the changes you want in the SQL Query window.

3 Choose OK.

Q+E executes the revised statement and displays the results in the Query window.

Defining a Query Using SQL

You can use SQL to define a query from scratch without opening a Query window or choosing any menu commands (although it is usually easiest to build a SQL SELECT statement using menu commands and then edit it if necessary).

ⅢⅢ Defining a query using SQL

You can build a SQL statement using any SQL clauses and expressions supported by the Q+E driver for the source database system. For more information, see the appendix for the database system.

1 Choose File Open.

2 In the Open dialog box, choose the SQL button.

3 Enter the SQL SELECT statement in the SQL Query window. (You can also enter any other SQL statement acceptable to the source database system.)

4 Choose OK.

Q+E executes the statement and displays the results in a Query window.

By default the SQL statement is sent to the current source database system. This is the source selected in the File Open dialog box.

If you want to select records from a different source, you can specify the source in the SELECT statement itself. To do this, you add a prefix to the first filename in the FROM clause.

For example, the following clause ensures that the source is dBASE:

FROM dBASEFile|C:\QE\EMP.DBF

To query this source	Add this prefix
dBASE	dBASEFile|
Text file	TextFile|
OS/2 SQL Server	SQLServer|
Oracle	Oracle|
OS/2 Extended Edition	EEDataMgr|
Microsoft Excel worksheet file	ExcelFile|

If the prefix is omitted, the statement is sent to the current source.

Part 3

Maintaining Database Files

Chapter 6   *Editing Records*

This chapter explains how to use Q+E to update and delete records, and to add new records to your database files. You do not have to save your changes when you are done. As soon as you change the data in a Query window, the underlying database files are automatically updated.

IMPORTANT   In some cases, you cannot edit records in the Query window. You cannot edit records if you installed a read-only version of Q+E or if the database file you want to edit is a read-only file. In addition, you cannot edit records if the Select Distinct command has been chosen. If you have joined database files in the Query window, editing is restricted. You can update values, but you cannot add new records or delete existing records. Other editing restrictions may apply to your database system. For more information, see the appendix for your database system.

Entering Edit Mode

When you first open a database file in a Query window, Q+E does not allow you to edit any records. This protects you from changing data accidentally as you select and format records. To change the values of fields, delete records, and add new records, you must choose the Edit Allow Editing command to enter edit mode. When you have made all your changes, you can leave edit mode to prevent any more changes.

Allowing editing

If your version of Q+E is read-only, the Edit Allow Editing command will not appear on the Edit menu. If editing is not allowed for the current database file or query, the Edit Allow Editing command is dimmed.

▶ In the Query window, choose Edit Allow Editing.

When editing is allowed, a check mark appears next to the command on the menu. Choose Edit Allow Editing again to turn off the command.

If you are in edit mode and you then use the Select Distinct command, the Edit Allow Editing command is automatically turned off.

NOTE   When you are in edit mode, Q+E does not automatically rerun the current query each time you update a record. To ensure that you are viewing the results of the current query, choose the Select Query Now command. For more information, see "Running the Current Query in Edit Mode" later in this chapter.

Editing Fields

To edit a field, you select the characters you want to replace and type the new values. You can select an entire field and replace everything in it, or you can select and change single characters within the field (for example, to correct a misspelling).

ⅠⅠⅠⅠⅠ   *Replacing the value in a field*

▷ Select the field and type the new value.

ⅠⅠⅠⅠⅠ   *Editing single characters within a field*

1 Drag to select the characters.

2 Type to replace the selection; press BACKSPACE or DELETE to delete the selection.

1 Select the field.

2 Press the LEFT ARROW or RIGHT ARROW key to position the cursor.

3 Press BACKSPACE or DELETE to delete characters; type to insert characters.

Before you move to a different field, you can undo your changes by pressing the ESC key or choosing the Edit Undo command. To move to a different field, press the TAB key or click the mouse.

Editing a Field in a Zoom Window

If a value is wider than its column, it may be more convenient to "zoom" the field and edit its value in a separate window. You must zoom to edit values with more than 255 characters.

ⅠⅠⅠⅠⅠ   *Zooming a field*

▶ Double-click the field. If you then double-click a different field, Q+E immediately displays its value in the zoom window.

▶ Select the field and choose Edit Zoom Field.

ⅠⅠⅠⅠⅠ   *Closing the zoom window*

▶ Click outside the zoom window, or choose the Close command from the zoom window's Control menu.

Updating Multiple Fields

If you want to update several fields with the same value, you can update them all at once.

ⅠⅠⅠⅠⅠ   *Updating multiple fields*

1 Select the fields you want to change.

2 Choose Edit Update All.

3 In the Value box, type the new value that you want to replace all the selected values.

4 Choose OK.

If you don't get the results you want, you can choose the Edit Undo command to undo your update.

Moving and Copying Data

You can move or copy values using the commands on the Edit menu. Since Q+E transfers data using the Windows or OS/2 Clipboard, you can move or copy data to a different database file as well as to another part of the same database file.

NOTE   You can also copy data to other Windows or OS/2 applications. For more information, see Chapter 8, "Transferring Data to Other Applications."

▪▪▪▪▪ Moving or copying the current selection

1 Select the data you want to move or copy. You can select characters in one field, the entire value in one field, or multiple fields.

2 Choose Edit Cut to delete the selection and place it on the Clipboard, or choose Edit Copy to copy the selection and place the copy on the Clipboard.

3 If you want to paste into a different database file, open the file.

4 Select the area where you want to paste the data from the Clipboard. If you are pasting multiple fields, select an area spanning the same number of columns and rows.

5 Choose Edit Paste to copy the values from the Clipboard into the selected area.

If you don't get the results you want, you can choose the Edit Undo command to undo the paste.

WARNING   Q+E only pastes into the selected area. If the paste area is smaller than the data you are pasting, the results are truncated. In addition, if the data you are pasting has a different data type than the column you are pasting into, Q+E may not be able to paste the data. For example, you cannot copy a name in the LAST_NAME column (Character data type) and paste it into the HIRE_DATE column (Date data type).

Editing Using a Form

Most database files have more columns than can be seen at one time in a Query window. It is often more convenient to edit records using Q+E's Form dialog box. In addition, the command buttons in the Form dialog box allow you to add, duplicate, and delete records.

ⅡⅡⅡ Editing records using a form

1 Select any field in a record you want to edit and choose Edit Form, or double-click the record number.

2 Edit record fields in the form. To move through the records, press the UP ARROW or DOWN ARROW key or use the scroll bar. To move between fields, press the TAB key. To cancel your changes to a record, choose the Restore button before you display a different record.

Any computed columns are dimmed, since they cannot be edited.

3 To exit the form, choose the Exit button.

ⅡⅡⅡ Adding a new record

▶ In the Form dialog box, choose the New button and then enter data in the new record's fields.

ⅡⅡⅡ Adding a duplicate record

▶ In the Form dialog box, display the record you want to duplicate and then choose the Dup button.

ⅡⅡⅡ Deleting a record

When you delete a record using a form, you cannot restore it using the Edit Undo command. If you want to be able to restore a deleted record, use the Edit Delete Records command.

▶ In the Form dialog box, display the record you want to delete and then choose the Delete button.

Customizing the Form Layout

You can specify a custom layout for the Form dialog box. By default, if there are more record fields than can fit vertically in the form, Q+E arranges the fields in two or more columns in the Form dialog box. If there are more fields than can fit in the dialog box, the last record fields are not displayed.

ⅡⅡⅡ Changing the form layout

1 Choose Edit Form Setup.

2 In the Number of Columns box, enter the maximum number of columns you want to appear in the Form dialog box.

3 In the Fields per Column box, enter the maximum number of fields you want to appear in any one column.

4 In the Field Width box, enter the maximum width in characters you want the edit box to be for each field.

Adding Records

You can use either the Edit Form command or the Edit Add Record command to add new records. Choosing Edit Add Record is the same as choosing Edit Form and then choosing the New button in the Form dialog box. You enter values in the blank fields of the form; you can add a duplicate record using the Dup button.

Edit Add Record is a quicker method if all you want to do is add a record. For more information on the Form dialog box, see "Editing Using a Form" above.

You can also add copies of existing records using the Edit Paste Append command. This command is useful if you want to copy a record and then modify it slightly to create a new, unique record.

///// Adding copies of existing records to a database file

When you paste records using the Edit Paste Append command, Q+E automatically adds them at the end of the database file.

1 Select all the fields in each record you want to copy.

   Click the record number.

   Select any field and press SHIFT+SPACEBAR.

2 Choose Edit Copy.

3 Choose Edit Paste Append. Q+E pastes the records at the end of the database file.

Deleting Records

You can delete one record at a time using a form. For more information, see "Editing Using a Form" above. You can also delete records directly using the Edit Delete Records command.

///// Deleting records without a form

1 Select a field in each record you want to delete.

2 Choose Edit Delete Records.

If you delete a record by mistake, you can restore it by choosing the Edit Undo command immediately after you choose Edit Delete Records.

Undoing Your Changes

You can use the Edit Undo command to undo the following changes:

- Editing the value in a field.
- Moving data to the Clipboard using the Edit Cut command.
- Pasting data from the Clipboard using the Edit Paste command.

- Updating fields using the Edit Update All command.
- Deleting records using the Edit Delete Records command.

You must choose Edit Undo immediately after you make the change. You can then choose Edit Undo again if you want to redo your change.

Running the Current Query in Edit Mode

When you are in edit mode, Q+E does not automatically resort and reselect the records if you edit columns that are part of the sort and selection conditions. Likewise, Q+E does not update a computed column if you change the values used in the computation.

If Q+E made these changes automatically, the screen could become very confusing. For example, as soon as you changed a value, a record might disappear if it no longer met the selection conditions. Or, a record might move to maintain the sort order.

If you add a new condition to the query, Q+E will automatically rerun the query. Otherwise, you must choose a command to specify that you want to resort, reselect, and recompute the records.

▐▐▐▐▐ Re-executing the current query

▶ Choose Select Query Now.

You can also use the Select Query Now command to see recent changes when you are accessing a shared database on a network. In this case, other users may be changing data while you are using Q+E. Select Query Now requeries the database, retrieving the most recent information.

Chapter 7   *Defining Database Files*

You can create new database files in Q+E's Define window. When you create a database file, you specify what fields you want it to contain, what type of data will be stored in each field, and the size of each field. After you define a new database file, you can open it in a Query window and enter data. You can also edit or delete an existing database definition. For example, you might delete a field from a database definition if you no longer need the data contained in the field.

This chapter provides general procedures for working with database definitions. Depending on your database system, there may be restrictions on what you can do with database definitions. Or, there may be additional ways to modify definitions, such as sorting and indexing in dBASE. For information specific to your database system, see the appendix for your database system.

NOTE   If you are using Q+E to access SQL Server, Oracle, or OS/2 Extended Edition, you define database *tables* instead of database files. In the main part of this manual, both database files and tables are referred to generically as database files.

Creating a New Database File

To create a new database file, you open a new Define window, define fields, and then save the definitions in a new file. After you have created the file, you can open it in a Query window and add records to the file.

IIIII   *Opening a blank Define window*

1   Choose File Define.

2   Select the source for the database file you want to define.

3   If options are available, choose the Options button, select the options you want, and choose OK.

4   Choose the New button.

Q+E displays a blank Define window.

IIIII   *Defining new fields in the Define window*

To define the fields in a new database file, you enter one row of information for each field in the Define window. The order of the field definitions determines the order of the columns when you open the database file in a Query window.

The information needed to define a field varies for different database systems. Q+E automatically displays a column in the Define window for each piece of information needed to define a field for the current source.

For detailed information on what parameters are required, see the appendix for your database system.

At a minimum, Q+E needs the following:

Column	Purpose
FIELD_NAME	Determines the default column head-ing for a field when you open a database file in a Query window.
TYPE	Determines the types of data that can be entered in the field. For example, if you specify a date/time data type, only date or time values can be entered in the field.
WIDTH	Determines the maximum number of characters that can be entered in the field.

1 In the FIELD_NAME column, type a name for the new field, and then press TAB to move to the TYPE column.

2 Select the type of field you want from the Data Types box, and then press TAB to move to the WIDTH column. You can select a field type by typing the first letter of the type.

3 Type the width you want, and then press TAB to move to the next column.

4 Fill in any other columns, and then press TAB to move to the next row.

5 Repeat steps 1 through 4 to add all the fields you want.

As you create a new database file, you can add, edit, or delete fields using the commands on the Edit menu. For more information, see "Editing a Database Definition" below.

ⅡⅡⅡ *Saving a new database file*

When you save a new database file, you are saving the field definitions—not actual data.

1 Choose File Save As.

2 Type a name for the database file. Q+E will add an extension for you automatically when appropriate.

3 If options are available, choose the Options button, select the options you want, and choose OK. Options depend on the type of database file. For more information, see the appendix for your database system.

4 Choose OK.

5 To close the Define window, choose File Close.

Now the database file exists, but it contains no records. To add records to the file, open it in a Query window, choose the Edit Allow Editing command, and enter data. For more information, see Chapter 6, "Editing Records."

Editing a Database Definition

You can modify the field definitions of an existing database file in a Define window.

There are restrictions on the types of changes that are allowed for some database systems. For example, you can open a SQL Server table in a Define window, but you must save any changes to a new table. For more information, see the appendix for your database system.

WARNING   If you make a field width smaller, some data may be truncated and lost when you save your changes. For example, if you change the width of the LAST_NAME field to 5, the name BENNETT is converted to BENNE because only 5 characters are allowed. If you change the type of a field, Q+E will try to convert the data to the new data type. If a value cannot be converted, the resulting field will be blank or NULL. If you delete a field, all the data in the field will be deleted when you save your changes.

Opening an existing database file in a Define window

1   Choose File Define.

2   Select the source for the database file you want to open.

3   Select the file you want to open.

4   Choose OK.

Q+E displays the field definitions for the database file.

Editing a field definition

▶   Select a value you want to change, and then type or select a new value.

You can also use the Edit Cut, Edit Copy, and Edit Paste commands to edit field definitions just as you use these commands to edit records in a Query window. For more information, see Chapter 6, "Editing Records."

Adding a field

When you add a new field definition, it can be inserted exactly where you want it.

1   Select any item in the row above or below where you want to insert the new definition.

2   Choose Edit Add After to open a blank row below the field you selected. Choose Edit Add Before to open a blank row above the field you selected.

3   Enter information to define the field.

You can also use the Edit Paste Append command to add a copy of an existing field definition just as you use this command to copy records in a Query window. For more information, see Chapter 6, "Editing Records."

ııııı   *Deleting a field*

When you remove a field in the Define window and save your changes, all the values stored in that column in the database file are deleted. Before you delete a field, make sure you no longer need all the data it contains.

1   Select any value in the row containing the field definition.

2   Choose Edit Delete Fields.

If you delete a field definition by mistake, you can restore it by choosing the Edit Undo command immediately after you choose Edit Delete Fields.

ııııı   *Saving edited field definitions to the same database file*

Depending on your changes, data may be deleted or truncated when you save. For example, if you deleted a field definition, all the data stored in that field will be deleted.

1   Choose File Save. Q+E asks you to confirm that you want to save your changes to the same database file.

2   Choose Yes.

If you want to discard your changes, you can use the File Close command to close the Define window without saving. Q+E will ask you to confirm that you do not want to save your changes.

ııııı   *Saving edited field definitions as a new database file*

If you save the edited definitions as a new file, Q+E copies the records from the old file to the new one, converting the data if you have changed the data types or widths of any fields.

1   Choose File Save As.

2   Type a new name in the Save As box.

3   If options are available, choose the Options button, select the options you want, and choose OK. Options depend on the type of database file. For more information, see the appendix for your database system.

4   Choose OK.

Deleting a Database File

To delete a database file and all its data, you delete the database file definition. Depending on the source, there may be restrictions on deleting database files. For more information, see the appendix for your database system.

ııııı   *Deleting a database file*

1   Choose File Define.

2   Select the source for the database file you want to delete.

3   In the File List box, select the file you want to delete. Depending on the source, this box may have a different name.

4   Choose the Delete button. Q+E asks you to confirm that you want to delete the file.

5   Choose Yes.

6   Choose OK to close the Define dialog box.

Printing a Database Definition

You can print a list of the field definitions for a database file.

///// Printing a database definition

1   Open the database file in a Define window.

2   Choose File Print, and then choose OK.

For more information on printing, see Chapter 4, "Saving a Query, Saving Results, and Printing."

Part 4

Using Q+E with Other Applications

Chapter 8   *Transferring Data to Other Applications*

This chapter explains how to transfer data from Q+E to other Microsoft Windows or OS/2 applications, such as Microsoft Excel and Microsoft Word for Windows. You can:

■ Copy data to the Clipboard and paste it into another application.

■ Copy data and *paste link* it into another application. This creates a dynamic link between the other application and Q+E, so whenever the database information in Q+E changes, the data in the application is updated automatically.

You can also use Microsoft Excel to extract records from an external database and paste them into a Microsoft Excel worksheet. For more information, see Chapter 9, "Using Microsoft Excel to Access Data on External Databases."

In addition, you can use macros in another application or program to control and exchange information with Q+E. For more information, see Chapter 10, "Using Dynamic Data Exchange to Communicate with Q+E."

Copying Data from Q+E to Another Application

When you do a simple copy and paste, you are taking a snapshot of the data in Q+E and pasting it into another application. The result is much like using a word processor to copy text from one document to another.

ııııı Copying data from Q+E to another application

1 Activate Q+E.

2 Select the data you want to copy to another application.

3 Choose Edit Copy.

4 Activate the other application, and open the file into which you want to paste.

5 Select the upper-left corner of the area where you want to paste the data.

6 Choose Edit Paste (or the equivalent command in the application you are using).

You can use the Edit Copy Special command instead of Edit Copy to control what is copied to the Clipboard. For example, you could copy column headings as well as values to the Clipboard. For more information, see "Using Edit Copy Special" later in this chapter.

Linking Another Application to Q+E

If an application supports Dynamic Data Exchange (DDE), you can create dynamic links between the application and Q+E. When you dynamically link to Q+E, data appears in the application just as if you had copied and pasted it. However, the data remains linked to the underlying database through Q+E. If the database information changes, the data in the application is updated automatically.

By linking to Q+E, you can guarantee that your worksheets and documents always contain up-to-date information. After you set up a link and save the worksheet or document, every time you open the file the link to Q+E is re-established and the most current database information is copied into the file.

The easiest way to create a link is to copy the Q+E data to the Clipboard and then paste it into the other application using the Edit Paste Link command. Both Microsoft Excel and Microsoft Word for Windows support Edit Paste Link. You can also create a DDE link by entering the linking formula directly in the application. For example, you can link a Microsoft Excel worksheet to Q+E by entering the linking formula in the formula bar.

This section explains how to:

❑ Link Microsoft Excel or Microsoft Word for Windows to a query file or Query window using Edit Paste Link.

❑ Link Microsoft Excel to a SQL statement using Edit Paste Link.

❑ Link Microsoft Excel to Q+E by entering the linking formula in the formula bar.

The linking procedures are similar for other Windows and OS/2 applications that support DDE. For information on linking in another application, see the documentation that came with the application.

Linking to a Query File or Query Window

You can link an application file to either a query (.QEF) file or the data in a Query window. Each time you open an application file linked to a query file, Q+E executes the commands saved in the query file and transfers the results to the application file. If Q+E is not running, the application will start it for you and open the query file. A link to a Query window, however, lasts only as long as the Query window is open. If you want to save a link, make sure you link to a query file, not a Query window.

⁗⁗ Linking a Microsoft Excel worksheet to a query file or Query window

1   Activate Q+E.

2   Use the commands in the Query window to display the information you want to link to another application. If you plan to save the link, make sure you save the results in a query (.QEF) file.

3   Select the information you want to link, and then choose Edit Copy to copy the data to the Clipboard. If you want to copy all the data in the Query window or specify other options, use Edit Copy Special. For more information, see "Using Edit Copy Special" later in this chapter.

4   Activate Microsoft Excel and open the worksheet into which you want to paste linked data. To complete this procedure, you must be using full menus.

5   Select the worksheet area where you want to paste the data. Make sure the area is large enough to contain all the data—any data that won't fit is not pasted.

6   Choose Edit Paste Link.

The Q+E data is linked to the worksheet. Any empty cells in the range you pasted into contain the value #N/A.

Note the external reference formula in the formula bar. This formula creates the actual link between the worksheet and Q+E.

For example, if you linked all the records in a query file named EMP.QEF to a Microsoft Excel worksheet, the external reference formula would look like this:

{=QE|EMP.QEF!ALL}

The following table explains the formula:

Item in formula	Description
QE	The application to which Microsoft Excel is linked.
|	The pipe character.
EMP.QEF	The document linked to. This could be a query file-name, as in this example, or a Query window name (for example, QUERY1).
ALL	The data linked to. In this example, all records in the query file were linked to the worksheet.

For more information on external reference formulas, see Chapter 8, "Working with Data from Multiple Documents," in the *Microsoft Excel User's Guide*.

▮▮▮▮▮ Linking a Microsoft Word for Windows document to a query file or Query window

1   Activate Q+E.

2   Use the commands in the Query window to display the information you want to link to another application. If you plan to save the link, make sure you save the results in a query (.QEF) file.

3   Select the information you want to link, and then choose Edit Copy to copy the data to the Clipboard. If you want to copy all the data in the Query window or specify other options, use Edit Copy Special. For more information, see "Using Edit Copy Special" later in this chapter.

4   Activate Microsoft Word for Windows and open the document into which you want to paste linked data.

5   Position the cursor where you want to paste the data.

6   Choose Edit Paste Link.

Linking Microsoft Excel to a SQL Statement

If you are linking a Microsoft Excel worksheet to Q+E, you can link directly to the SQL SELECT statement for a Query window or query file. When you link to the SQL statement, the SELECT statement appears in the external reference formula. Then, if you want to change your query, you can revise the SQL statement directly in the formula bar instead of reactivating Q+E.

For an introduction to SQL, see Chapter 5, "Editing the Current SQL Statement."

ιιιιι Linking a Microsoft Excel worksheet to a SQL statement

1   Activate Q+E.

2   Use the commands in the Query window to construct the SQL statement to which you want to link. If you plan to save the link, make sure you save the results in a query (.QEF) file.

3   Choose Edit Copy Special.

4   Under Link Formula, select SQL Text.

5   Select any other options you want in the dialog box. For more information, see "Using Edit Copy Special" below.

6   Choose OK.

7   Activate Microsoft Excel and open the worksheet into which you want to paste linked data.

8   If necessary, choose Options Full Menus to display full menus.

9   Select the rectangular area you want to contain the data.

10 Choose Edit Paste Link.

The Q+E data is linked to the worksheet. Note that the SQL SELECT statement appears as part of the external reference formula in the formula bar.

For example, if you linked to a SQL statement that retrieved the names and salaries of all employees in EMP.DBF making at least $40,000, the external reference formula would look like this:

{=QE|'SELECT FIRST_NAME,LAST_NAME,SALARY FROM C:\QE\EMP.DBF WHERE SALARY >= 40000'!ALL}

ιιιιι Editing the SQL statement

You can change the SQL SELECT statement in the external reference formula and then re-execute the revised statement to display the new results in your worksheet. You can also revise a formula that links to a Query window or query file, although to modify these types of linking formulas, it is best to reactivate Q+E and change the query in the Query window.

1   Select any worksheet cell that contains results from the current SQL SELECT statement.

2   Edit the SQL statement in the formula bar. For example, you could delete a field name from the statement, or add or change selection conditions.

3   Press CTRL+SHIFT+ENTER.

Q+E displays the results of the revised SQL statement in your Microsoft Excel worksheet. If you reactivated Q+E, you would see that Q+E created a new Query window to correspond to the new SQL statement.

Creating a Link in the Microsoft Excel Formula Bar

If you want to skip the commands in the Query window altogether, you can enter an external reference formula in the Microsoft Excel formula bar and then display the results in the worksheet.

▥▥▥   Linking to Q+E in the Microsoft Excel formula bar

1   Activate Microsoft Excel and open the worksheet you want to link to Q+E.

2   Select the worksheet cells you want to contain the linked data.

3   Select the formula bar, and then type the external reference formula. The formula could specify a link to a Query window, query file, or SQL statement.

4   Press CTRL+SHIFT+ENTER.

The selected records are retrieved from Q+E and pasted into the worksheet.

Solutions to Common Errors When Linking Microsoft Excel to Q+E

This section provides solutions to a few of the common problems encountered when linking Microsoft Excel worksheets to Q+E.

Q+E returns #NULL!

If the amount of data to be transferred exceeds the DDE limit of 64K bytes or if your system is low on memory, Q+E will not be able to return all the requested data. In this case, Q+E returns the #NULL! error value.

To solve this problem, close any other applications you have running and try again. If the #NULL! error value is still returned, the DDE limit has been exceeded. You can either change the Q+E query to return fewer records or use an area specification in a Microsoft Excel macro to return the results in multiple parts. For information on defining an area specification, see Chapter 10, "Using Dynamic Data Exchange to Communicate with Q+E."

Q+E cannot be started

If Q+E is not running and you open a worksheet that is linked to Q+E, Microsoft Excel will ask if Q+E should be started. If you tell it to start Q+E but no data is returned, Microsoft Excel couldn't find the QE.EXE program file.

To solve this problem, make sure your AUTOEXEC.BAT file contains the path to QE.EXE so that the system can find it.

Q+E returns #NAME?

If there is an error in the linking formula, Q+E returns the #NAME? error value. For example, if you open a worksheet that is linked to Q+E and a file referred to in the linking formula has been deleted, Q+E will fill the array in the worksheet with the #NAME? error value.

To solve this problem:

- Make sure all files that are referred to in the formula really exist.
- Use full pathnames in the formula to ensure that the files are found.
- If the linking formula contains a SQL statement, make sure it is correct.

If necessary, you can generate the linking formula again by defining a query in Q+E and paste linking it into Microsoft Excel.

Using Edit Copy Special

You can use the Edit Copy Special command instead of Edit Copy to specify additional options when you copy data to the Clipboard. You must use Edit Copy Special if you want to link to a SQL statement when you paste link into a Microsoft Excel worksheet.

When you choose Edit Copy Special, a dialog box appears in which you can select the following options:

- **Include Column Headings** Turn on this check box if you want to copy the column headings, as well as the records, to the Clipboard.
- **Include Record Numbers** Turn on this check box if you want to copy the record numbers to the left of each record, as well as the records, to the Clipboard.
- **Link Formula** If you want to paste linked data into a Microsoft Excel worksheet, select the style of formula you want to appear in the formula bar:
- **SQL Text for Excel 2.x** Turn on this check box if you want to copy data from Excel version 2.x.

Select	To
Query Name	Include the name of the Query window or query file in the formula. For example: =QE|EMP.QEF!ALL
SQL Text	Include the SQL SELECT statement in the formula. For example: =QE|'SELECT LAST_NAME,FIRST_NAME FROM EMP'!ALL

■ **Link Data** Select the data you want to copy to the Clipboard:

Select	To
Entire Window	Copy every record in the Query window.
Selected Area	Copy the selected data in the Query window.
SQL Text	Copy the current SQL SELECT statement to the Clipboard. When you paste, you will receive the SELECT statement, not the resulting records.

Chapter 9   Using Microsoft Excel to Access Data on External Databases

This chapter explains how to use Microsoft Excel to access data on external data-bases and transfer query results to Microsoft Excel worksheets. To do this, you must first open the Microsoft Excel add-in macro sheet named QE.XLA. Then you can use Microsoft Excel database commands to extract records from an external database in much the same way that you extract records from an internal Microsoft Excel database.

You can also use Microsoft Excel macros to control and exchange information with Q+E. For more information, see Chapter 10, "Using Dynamic Data Exchange to Communicate with Q+E."

Opening the QE.XLA Add-in Macro Sheet

The add-in macro sheet QE.XLA modifies the Microsoft Excel Data menu so that you can use Data commands to extract records from external database files.

▮▮▮▮▮   Opening the QE.XLA macro sheet

1   Activate Microsoft Excel.

2   Choose File Open.

3   Move to the directory that contains the QE.XLA file. By default, this is the subdirectory QEMACRO in the directory XLSTART.

4   In the Files box, select QE.XLA, and then choose OK.

When you install Q+E, the Setup program creates a subdirectory named QEMACRO in the Microsoft Excel startup directory (XLSTART), and then copies QE.XLA and another macro named QESTART.XLA into QEMACRO. If you want the external database commands to appear on the Data menu every time you start Microsoft Excel, move QESTART.XLA into the XLSTART directory. If you choose an external database command, Microsoft Excel will then load QE.XLA. This minimizes the startup time for launching Microsoft Excel, but makes the external database commands readily available if you need them. If you prefer, you can move QE.XLA into XLSTART and delete QESTART.XLA. For more information on add-in macros, see the *Microsoft Excel User's Guide*.

Changes to the Microsoft Excel Data Menu

After you open the QE.XLA macro sheet, three additional commands appear on the Microsoft Excel Data menu—Data Paste Fieldnames, Data SQL Query, and Activate Q+E. In addition, existing Data menu commands are modified so that they work on external as well as internal databases.

The following table summarizes how you can use the Data menu commands to access external data. See the remainder of this chapter for detailed information on connecting to an external database and extracting external data.

Use this command	To
Data Set Database	Connect to the database files from which you want to extract data.
Data Paste Fieldnames	Select field names from the database files to use in your criteria range and extract range.
Data Set Criteria	Define the criteria range.
Data Set Extract	Define the extract range. You can also simply select the extract range before choosing Data Extract.
Data Extract	Extract the data that meets the specified criteria.
Data Delete	Delete the data that meets the specified criteria.
Data SQL Query	Extract the results of a SQL SELECT statement.
Data Activate Q+E	Activate the Q+E program.

Connecting to an External Database File

To extract records from an external database file, you must first connect to the database file using the Data Set Database command. Normally, you define a worksheet selection as the database range. When QE.XLA is open, you can use the Data Set Database command to define an internal database range (as usual) or select one or more external database files.

After you have connected to an external database file, you can use Data Set Database to:

■ Switch back and forth between an internal database range and the external database file. The Data menu commands will operate on whichever is active.

■ Change the database file or add another external database file. For example, you might connect to another database file so that you can create joins between files.

ⅢⅢ Connecting to an external database file

1 Choose Data Set Database.

If there are database files that were previously connected with Data Set Database, Q+E lists the source names and pathnames in the Set Database dialog box. In this case, skip this procedure and see the procedures below for information on switching between and changing or adding to existing connections.

2 Select the External Database button and choose OK.

3 In the Source box, select the source for the database file you want to access. The Source box lists the database systems for which drivers have been installed and any computer servers that have been logged on to.

If the source for the file you want to access does not appear in the list, choose the Sources button to log on to the source. For more information, see "Logging On to or Off from an External Source" below.

4   In the Directory List box, select the directory that contains the file you want to access and choose OK. Depending on the source, this box may have a different name.

5   In the File List box, select the database file you want to access. This box lists all files for the current source that are in the current directory. Depending on the source, this box may have a different name.

6   To specify additional options, choose the Options button, select the options you want, and choose OK.

7   Choose OK to return to the Set Database dialog box. Note that the dialog box now lists the file you connected to and includes Change and Add buttons.

If you want to connect to another database file (for example, to specify a join), choose the Add button and repeat steps 4 through 7. You can only connect to multiple database files from the same source.

If you are done connecting to files, choose OK.

Switching between an internal database and the current external database file(s)

Once you have selected an external database file, you can use the Data Set Database command to specify whether you want the Data menu commands to operate on an internal database range or on the current external database file(s). Microsoft Excel will keep track of both so that you can easily switch back and forth between them.

1   Choose Data Set Database.

The Set Database dialog box lists the current external database file(s).

2   Select the Current Selection button if you want Microsoft Excel to name the current worksheet selection "Database" and all Data menu commands to operate on this database range.

Select the External Database button if you want the Data menu commands to operate on the current external database file(s).

3   Choose OK.

For information on changing or adding to the current external database files, see the next procedure.

Changing or adding to the current external database files

1   Choose Data Set Database.

The Set Database dialog box lists the current external database files.

2   Select the External Database button.

3   Choose the Change button if you want to remove the current connections and select a different database file.

Choose the Add button if you want to add another database file to the current external database files.

4 If you chose the Change button, select a different source in the Source box, if necessary. If the source for the file you want to access does not appear in the list, choose the Sources button to log on to the source. For more information, see "Logging On to or Off from an External Source" below.

5 In the Directories box, select the directory that contains the file you want to access and choose OK. Depending on the source, this box may have a different name.

6 In the Files box, select the database file you want to access. This box lists all files for the current source that are in the current directory. Depending on the source, this box may have a different name.

7 To specify additional options, choose the Options button, select the options you want, and choose OK.

8 Choose OK to return to the Set Database dialog box.

   If you want to add another database file, choose the Add button and repeat steps 5 through 8.

   If you are done, choose OK.

Logging On to or Off from an External Source

If the source you want to access is not available in the Set External Database dialog box, you can choose the Sources button to log on to the source. You can also log off from a source to free up resources while you are using Microsoft Excel.

⚏ Logging on to a source from the Set External Database dialog box

1 In the Set External Database dialog box, choose the Sources button.

2 In the Available Sources box, select the source you want to log on to. This box lists the sources for which drivers have been installed and that must be logged on to (for example, SQL Server but not dBASE).

3 Choose the Logon button.

4 Enter or select the information necessary to log on to the source and then choose OK. For more information, see the appendix for the source.

5 Choose Close to return to the Set External Database dialog box.

Now you can select the source you want to access in the Source box.

⚏ Logging off from a source from the Set External Database dialog box

1 In the Set External Database dialog box, choose the Sources button.

2 In the Current Connections box, select the source you want to log off from. This box lists the computer servers that have been logged on to.

3 Choose the Logoff button.

4 Choose Close to return to the Set External Database dialog box.

Selecting Field Names from the External Database File

After you connect to an external database file, you select the fields to use in your criteria range and in your extract range. For more information on criteria and extract ranges, see Chapter 10, "Analyzing and Reporting Database Information," in the *Microsoft Excel User's Guide*.

You use the Data Paste Fieldnames command to select the field names. The field names are pasted into your worksheet, starting with the active cell and continuing to the right until all the field names are pasted.

If you are connected to more than one external database file, the names of the fields are a combination of the name of the database file that contains the field and the field name, in the format *database_file.fieldname*. For example, a field named "EMP.DEPT_ID" tells you that the information you extract will come from the DEPT_ID field in the database file named EMP. Otherwise, only the field names are displayed.

NOTE   You can also use the Data Paste Fieldnames command to set up criteria and extract ranges for an internal database range. If a Microsoft Excel database is active, the Data Paste Fieldnames dialog box contains the labels from the first row of the range named "Database".

ııııı Selecting field names from the external database file

You can select field names and paste them into the worksheet in the order they appear in the database file. Or, you can order them differently using the Order Fields option.

1 Select the cell where you want to start pasting field names.

2 Choose Data Paste Fieldnames. The Fields box lists all the fields in the current external database file. If there is more than one external database file, the field names are grouped by filename.

3 If you want to paste all the field names in the order they appear in the Fields box, choose the Paste All button and skip the rest of this procedure.

4 If you want to paste only some of the field names, but still in the order they appear in the Fields box, select the names in the Fields box, choose the Paste button, and skip the rest of this procedure.

To select more than one field name with the mouse, hold down SHIFT and click additional field names.

To select more than one field name with the keyboard, hold down CTRL, use the ARROW keys to select each additional field, and then press SPACEBAR.

5 If you want to paste field names in a different order than they appear in the Fields box, choose the Order Fields button. The dialog box expands.

6 Add field names to the Selected Fields box in the order you want to paste them, and then choose the Paste button. The following table describes your options for building the list of field names:

To	Do this
Add one or more field names to the list.	Select the field name(s) in the Available Fields box, and choose the Add button.
Insert one or more field names in the list.	Select the field name(s) in the Available Fields box, select the field name just below where you want to insert the field name(s) in the Selected Fields box, and then choose the Add button.
Add all fields to the list.	Choose the Select All button, and then choose the Add button. This option is useful if you want to select all field names except for a few. After you choose Select All, you can deselect field names in the Available Fields box.
Remove a field name from the list.	Select the field name in the Selected Fields box, and choose the Remove button.
Clear the list.	Choose the Clear All button.

If you attempt to paste more field names than the worksheet can accept, Microsoft Excel will paste as many as it can and then tell you that it could not paste all the field names.

If Q+E is not running, you will be alerted that the field names are not available. In this case, reconnect to the external database files and try again.

Extracting Data from the External Database Files

Before you can extract data, you need to tell Microsoft Excel what to extract by defining the criteria range and the extract range.

To define the criteria range:

■ Enter the criteria beneath the field names you have listed in your worksheet.

■ Define the criteria range by selecting the field names and the criteria entered below them.

■ Choose Data Set Criteria.

NOTE   If you do not define a criteria range, Microsoft Excel will extract all the records in the external database files.

To identify the extract range:

▣ Define the extract range by selecting the field names that you want to extract. You can also use the Data Set Extract command to define the extract range.

▣ Choose Data Extract to extract the data.

You can also define a criteria range only and then use the Data Delete command to delete the external data that meets the specified criteria.

Entering the Criteria

When you enter criteria, you are telling Microsoft Excel which specific records you want to extract from the current external database file. You type your criterion beneath the appropriate field name. For example, you could enter a criterion to extract employee records for those employees whose salary is greater than $40,000.

Field name

```
SALARY
>40000
```

Criterion   *Criteria range*
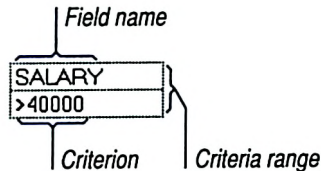
If there is more than one current external database file, the field name is prefaced by the database filename:

Database file name

Field name

```
EMP.SALARY
>40000
```
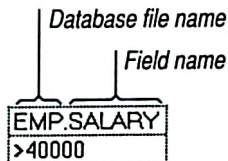
You can use the following arithmetic operators when entering criteria:

Addition	+
Subtraction	−
Multiplication	*
Division	/
Equal	=
Not equal to	<> or !=
Less than	<
Less than or equal to	<=
Greater than	>
Greater than or equal to	>=

Computed Criteria

The way you enter computed criteria to query a database that resides within Microsoft Excel is somewhat different from the way you enter computed criteria to query an external database file. When you query an external database file, you can use SQL-like commands, which are different from Microsoft Excel commands.

For example, suppose you want to extract the names of employees who have been hired within the last 30 days.

To query a Microsoft Excel database, you enter:

This field name cannot be a field name
from the table you are querying.

| Date Hired |
| HIRE_DATE < NOW() - 30 |

Criterion

To query an external database file using SQL, you enter:

This field name should reference the field
name you are querying.

| HIRE_DATE |
| < DATE() - 30 |

Criterion

In a computed criterion, you can include a formula referencing more than one field.

When you specify criteria to query an external database file, you cannot use Microsoft Excel computed criteria because they are Microsoft Excel-specific capabilities that cannot be translated into SQL-equivalent functions. For example, queries created and applied against external database files cannot use Microsoft Excel cell references or specific Microsoft Excel functions such as "=AVERAGE()".

Joining External Database Files

Sometimes, you may want to cross-reference information from one database file and use it to select records from another database file.

You can combine database files by including in your criteria fields that contain the same data in both database files. To do so, you need to tell Microsoft Excel to cross-reference this information. For example, to join EMP.DBF and DEPT.DBF based on the common values in their ID fields, you'd use the following format:

Database files to join

```
 _____
| [DEPT.MGR_ID ] |
| [=EMP.EMP_ID ] |
|_____|
```

Names of fields that
contain data in common
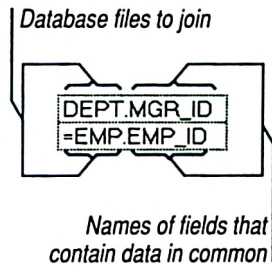
DEPT.DBF contains a record of information about each department. EMP.DBF contains a record for each employee. You want a report showing the employee record for each department manager. The criteria above can be translated as follows: For every manager ID number in DEPT.DBF, find the corresponding ID number and record in EMP.DBF.

When you enter criteria that join two database files and then perform an extract, the extracted data looks as if it came from a single database file.

Defining the Extract Range

When you define the extract range, you indicate which fields of information you want and where to put the information. The appropriate field names must be entered on your worksheet. You can either type them or use Data Paste Fieldnames. You then select the field names, or define the extract range using Data Set Extract and then choose Data Extract. If the extract range includes only the field names, Microsoft Excel places the data below the field names, replacing any information in the cells. If the extract range includes the field names and a range of cells beneath these field names, Microsoft Excel fills only that range.

The extract range must be separate from the criteria range. The extract range can contain the same fields as those in the criteria range, or it can contain a different set of fields from the same database file.

Extracting the Data

To extract the data, choose the Data Extract command. Microsoft Excel uses the field names and criteria you specify to query the current external database files and extract the information into Microsoft Excel.

When you extract the records, you can copy them into the worksheet, or you can copy them and maintain a link to the source database file. Linked extracts remain linked until you close the Q+E or Microsoft Excel window.

ⅢⅢ Initiating a query

1   Enter the criteria beneath the field names.

2   Select the criteria range.

3   Choose Data Set Criteria.

4 Define the extract range by selecting either the field names only or a range of cells including the field names. You can also use the Data Set Extract command to define the extract range.

5 Choose Data Extract. When Microsoft Excel receives the records, it tells you how many records were received and gives you choices for saving the data.

6 To place the information in the worksheet, select the Linked or Unlinked button to specify the type of paste operation and then choose the Paste button.

To save the information in a file, choose the Save As button. Microsoft Excel will display the File Save As dialog box so you can name the file.

To cancel, choose Cancel.

When the data is in Microsoft Excel, you can use Microsoft Excel's statistical functions to analyze it.

If you see a message that the criteria are invalid, review your criteria and be sure you chose Data Set Criteria to identify the criteria range. For more information, see "Entering the Criteria" earlier in this chapter.

If you see a message that the extract range is invalid, review the extract range and be sure you selected the field names or defined the extract range using Data Set Extract before choosing Data Extract. For more information, see "Defining the Extract Range" earlier in this chapter.

Deleting External Data

You can also delete external data that meets the specified criteria using the Data Delete command.

Deleting external data

1 Define the criteria range.

2 Choose Data Delete. Microsoft Excel asks you to confirm that you want to delete the data.

3 Choose OK.

Executing a SQL Query

Up to now, this chapter has explained how to retrieve external database records using Microsoft Excel's regular database functions. You can also use the Data SQL Query command to enter or edit a SQL SELECT statement and retrieve the results of the query into the current worksheet.

When you set up criteria and extract external database records, Microsoft Excel builds the corresponding SQL SELECT statement, just like Q+E does when you choose commands in a Query window. You can use Data SQL Query to edit this statement or save it to a query (.QEF) file. You can also open an existing query file and display the results in the worksheet, or you can define a new SQL query from scratch.

For an introduction to SQL queries, see Chapter 5, "Editing the Current SQL Statement." For information about query (.QEF) files, see Chapter 4, "Saving a Query, Saving Results, and Printing."

ⅲⅲⅲⅲ *Extracting the results of a SQL query*

1   Select the worksheet cell where you want to begin pasting the results. Select a range to limit the number of records and columns that are returned.

2   Choose Data SQL Query. The SQL Query dialog box appears.

If you have already extracted external database records in the current session, you see the SQL SELECT statement that describes the query.

3   You can edit and save the current query, open another query, create a new query, or run a query. The following table describes your options in the SQL Query dialog box.

To	Do this
Change the current SQL query.	Edit the SQL statement in the dialog box.
Save the current SQL query to a file.	Choose the Save button. You can save the query to a query (.QEF) file in the Save dialog box.
Display an existing SQL query saved in a file.	Choose the Open button. You can open any text file or query (.QEF) file in the Open dialog box. If there is an unsaved query in the SQL Query dialog box, you are prompted to save it before it is cleared.
Clear the current SQL query so that you can enter a new one.	Choose the New button. If the current query has not been saved, you are prompted to save it before it is cleared. You can enter any SQL statement that is supported by the current source. For more information, see the appendix for the source.
Execute the current SQL query.	Choose the Run button. While the query is running, the status appears in the status bar.
Cancel the Data SQL Query command	Choose the Cancel button.

When you run a query from the Data SQL Query dialog box, Microsoft Excel tells you the number of lines received. You then have the choice of pasting the data on the worksheet linked or unlinked, saving the data to a file, or canceling the query, just as you do when you use the Data Extract command.

Activating Q+E from Microsoft Excel

When the QE.XLA macro sheet is open, you can quickly move from Microsoft Excel to Q+E whenever you need to adjust a query directly in a Query window.

ııııı   *Activating Q+E from Microsoft Excel*

▶   Choose Data Activate Q+E.

Q+E becomes the active window. If Q+E is not running, Microsoft Excel starts it.

Using Macro Functions with Q+E

The QE.XLA macro sheet includes six macro functions you can use. Since QE.XLA is an add-in macro, the functions appear in the Formula Paste Function dialog box just like built-in Microsoft Excel macro functions.

This section follows the rules for syntax and arguments used in the *Microsoft Excel Function Reference*. The following list summarizes the functions:

DB.EXTRACT*(unique,destination,filename,linked)*

DB.GET.DATABASE*(info_type)*

DB.LOGON*(source,database,username,password)*

DB.PASTE.FIELDNAMES*(filename)*

DB.SET.DATABASE*(internal,source,filename)*

DB.SQL.QUERY*(type,query,destination,filename,linked)*

All except DB.GET.DATABASE are equivalent to commands on the Data menu. The command equivalent functions each have an alternative "?" form, such as DB.EXTRACT?( ). This form of the function displays the appropriate dialog boxes while the macro is running. Each dialog box remains on the screen until you make the selection(s) you want.

NOTE   QE.XLA macro functions can be recorded just like built-in macro functions. When you enter or record a QE.XLA macro function, the path to QE.XLA is automatically saved with the function. If you move QE.XLA to a different directory, you may need to re-enter the QE.XLA macro functions so that Microsoft Excel can find them. This problem will not occur if you keep your macro sheets in the same directory as QE.XLA, or you place them in the Microsoft Excel startup directory (XLSTART).

DB.EXTRACT*(unique,destination,filename,linked)*
DB.EXTRACT?*()*

Equivalent to the Data Extract command when DB.XLA is open and external database files are active. Extracts the external records that match the defined criteria.

Unique   is a logical value specifying whether duplicate records are extracted when the source is SQLServer, Oracle, or EEDataMgr. If *unique* is TRUE, Microsoft Excel extracts unique records only; if FALSE, extracts all records matching the defined criteria. If omitted, Microsoft Excel assumes FALSE and extracts all records matching the defined criteria. This argument has no effect when the source is dBASE, TextFile, or ExcelFile.

Destination is a number from 1 to 3 specifying the destination for the extracted records.

Destination	Description
1	Extracts information to the selected range. This is the default.
2	Extracts information to a file named *filename*.
3	Extracts information to the selected range *and* saves it to a file named *filename*.

Filename is the name of the file in which to save the information if *destination* is 2 or 3. If omitted, the default is "QE mm/dd/yy hh:mm:ss." If *destination* is 1, *filename* is ignored and can be omitted.

Linked is a logical operator specifying whether the extracted information is linked to the source database files. If *linked* is TRUE, the data is linked; if FALSE, it is not.

DB.GET.DATABASE(info_type)

Returns the requested information regarding the active database.

Info_type is a number from 1 to 8 specifying what type of information will be returned.

Info_type	Returns
1	1 if internal database range is active; 2 if external database files are active.
2	Name of the current source.
3	Horizontal array containing the locations of the external database files or tables that are connected to Q+E.
4	Horizontal array containing the names of the external database files or tables that are connected to Q+E.
5	Schema name in File Open dialog box (such as Directory or Owner).
6	Filename in File Open dialog box (such as File or Table).
7	Horizontal array containing the names of the sources that have been logged on to.
8	Horizontal array containing the names of the sources that can be logged on to.

DB.LOGON(source,database,username,password)
DB.LOGON?()

Equivalent to choosing the Sources button in the Set External Database dialog box and entering information in the Source Connections dialog box. Logs on to a computer server that contains database tables from which you want to extract data.

Source specifies the source to log on to. It can be any source listed in the Source Connections dialog box.

Database is the name of the database that contains the tables you want to access.

Username is your log in user name.

Password is your user password.

DB.PASTE.FIELDNAMES*(filename)*
DB.PASTE.FIELDNAMES?*()*

Similar to the Data Paste Fieldnames command. Inserts all the field names for the given database file. To insert selected field names from a database file, use the Microsoft Excel built-in FORMULA or FORMULA.ARRAY function.

Filename is the name of the database file containing the field names you want to paste. To specify more than one database file, enter an array—for example, {"emp.dbf","dept.dbf"}.

DB.SET.DATABASE*(internal,source,filename)*
DB.SET.DATABASE?*()*

Equivalent to the Data Set Database command when the QE.XLA macro is open. Connects Microsoft Excel to one or more external database files from which you want to extract data; defines the active database as an internal database range or the current external database files.

Internal is a logical value specifying whether the Data menu commands will operate on the current worksheet selection or the external database files specified by *filename*. If *internal* is TRUE, the internal database is active; if FALSE, the external database files are active.

Source is the database system that contains the external database files you want to access.

Filename is the name of the external database file you want to access. To specify more than one database file, enter an array—for example, {"emp.dbf","dept.dbf"}.

DB.SQL.QUERY*(type,query,destination,filename,linked)*
DB.SQL.QUERY?*()*

Equivalent to the Data SQL Query command. Executes a SQL query and transfers the results to Microsoft Excel.

Type is a number from 1 to 3 specifying the type of query to execute.

Type	Result
1	Executes previous query; error if no previous query in this session. This is the default.
2	Executes query defined by *query*, where *query* is a text string.
3	Executes query defined by *query*, where *query* is a filename.

Query specifies the query to execute. If *type* is 2, *query* is a text string of up to 256 characters specifying a SQL query supported by Q+E. If *type* is 3, *query* is the name of a file containing a valid SQL query. This argument is ignored if *type* is 1.

Destination is a number from 1 to 3 specifying the destination for the extracted records.

Destination	Description
1	Extracts information to the selected range. This is the default.
2	Extracts information to a file named *filename*.
3	Extracts information to the selected range and saves it to a file named *filename*.

Filename is the name of the file in which to save the information if *destination* is 2 or 3. If omitted, the default is "QE mm/dd/yy hh:mm:ss." If *destination* is 1, *filename* is ignored and can be omitted.

Linked is a logical operator specifying whether the extracted information is linked to the source database files. If *linked* is TRUE, the data is linked; if FALSE, it is not.

Chapter 10 Using Dynamic Data Exchange to Communicate with Q+E

This chapter explains how to use Dynamic Data Exchange (DDE) in macro languages and programming languages to communicate with Q+E and send information back and forth. For example, you can use DDE in Microsoft Excel macros to:

□ Start Q+E.

□ Send information to Q+E.

□ Get data from Q+E.

□ Carry out any Q+E command.

This chapter first describes the DDE functions and parameters you can use to communicate with Q+E. It then provides examples of Microsoft Excel macros that use DDE to exchange information with Q+E. For specific information on entering DDE function statements, see the documentation for your macro or programming language.

For details on the commands that you can send to Q+E via DDE, see Chapter 11, "Q+E Execute Command Reference."

Using DDE

To use DDE to communicate with Q+E, a macro or program must perform these steps:

□ Open a channel. All DDE communication occurs on a channel. You use the INITIATE function to open a channel from another application to Q+E. The application that opens the channel (for example, Microsoft Excel) is called the *client*; Q+E is the *server*.

■ Send or request data. Once a channel is opened, the client application and Q+E can exchange data. You use the REQUEST function to request data from Q+E and the POKE function to send data to Q+E. You can also use the EXECUTE function to execute Q+E commands.

■ Close the channel. After the client and Q+E have exchanged data, you use the TERMINATE function to close the channel between the applications.

The following example demonstrates a Microsoft Excel macro that uses DDE to exchange information with Q+E. The functions and their parameters are explained in detail later in this chapter.

```
chan=INITIATE("QE","SELECT LAST_NAME, HIRE_DATE FROM EMP.DBF")
name=REQUEST(chan,"R1C1")
=EXECUTE(chan,"[ALLOW.EDIT(TRUE)]")
=POKE(chan,"R1C1","R10C10")
=TERMINATE(chan)
```

The INITIATE function opens a DDE channel with Microsoft Excel as the client and Q+E as the server. The channel is linked to a Query window containing the results of the SQL SELECT statement. The channel number is saved in the variable *chan*.

The REQUEST function asks for the value in row 1, column 1 of the Query window. The value in the LAST_NAME field of the first employee record is returned and saved in the variable *name*.

The EXECUTE function sends the Edit Allow Editing command to Q+E, permitting updates to the data in the Query window.

The POKE command changes the value of the LAST_NAME field in row 1, column 1 of the Query window to the value stored in R10:C10 of the macro sheet.

The TERMINATE function closes the DDE channel.

Macro languages differ from one application to another. For example, the same macro in Microsoft Word for Windows looks like this:

```
chan=DDEInitiate("QE","SELECT LAST_NAME, HIRE_DATE FROM EMP.DBF")
name$=DDERequest$(chan,"R1C1")
DDEExecute chan,"[ALLOW.EDIT(TRUE)]"
DDEPoke chan,"R1C1","Jones"
DDETerminate chan
```

In this case, the POKE command changes the LAST_NAME value in row 1, column 1 to Jones.

The following sections describe each DDE function in detail.

Opening a DDE Channel

To open a channel to Q+E, you use the INITIATE function. This function has two parameters:

Parameter	Description
app_name	The name of the server application. When you initiate a channel to Q+E, this parameter is "QE".
topic	The part of the server application that you want to exchange information with. This can be a Query window or the Q+E application itself.

For example, to open a channel from a Microsoft Excel macro sheet to a Query window named Query1, the INITIATE function might look like this:

```
chan=INITIATE("QE","Query1")
```

The INITIATE function returns a channel number, which is used as a parameter to all subsequent DDE functions called on that channel. In the example above, the channel number is saved in the variable *chan*. You can call the INITIATE function multiple times to open multiple channels.

You can link to the following Q+E topics:

Topic	Result
Query window name	Links to the Query window. Q+E must already be running, and the Query window must be open. For example: =INITIATE("QE","QUERY1")
Query file name	Links to the Query window that displays the results of the query file. Q+E opens the query file if it is not already open. For example: INITIATE("QE","EMP.QEF")
SQL SELECT Statement	Q+E executes the SQL SELECT statement and displays the results in a Query window. The channel is then linked to this Query window. For example: =INITIATE("QE","SELECT * FROM EMP")
SYSTEM	Links to the Q+E application instead of a single Query window. This allows you to open and control multiple Query windows on one DDE channel. For example: =INITIATE("QE","SYSTEM")

Except when the topic is a Query window name, the INITIATE function automatically starts Q+E if it is not already running.

An error is returned if Q+E is not running and cannot be started, or if the topic is not valid.

Requesting Information

Once you have opened a channel to Q+E, you can get data from Q+E using the REQUEST function. This function has two parameters:

Parameter	Description
channel_num	The value returned by the INITIATE function.
item	The data to be returned. This can be the contents of a field or range of fields in a Query window, or other types of information, such as the number of records in the Query window.

For example, the following Microsoft Excel macro statement retrieves the data in row 3, column 5 of the Query window and places it in the variable *date*.

date=REQUEST(chan,"R3C5")

The results can then be copied from the macro to a worksheet or document. For information on returning data directly to a worksheet or document, see "FETCH and FETCH.ADVISE" under "Executing Commands" later in this chapter.

An error is returned if either the channel number or the item is invalid.

Retrieving the Data in a Field or Range of Fields

You use an *area specification* item (such as "R3C5" in the example above) to retrieve the data in a specified field or range of fields in a Query window. You can use the area specification ALL to retrieve all the data in the Query window. Or, you can use the $RnCm:RxCy$ notation (as in Microsoft Excel) to retrieve part of the data in the Query window. For example:

This item	Returns
ALL	All the data in the Query window.
R3C5	The value in row 3, column 5.
R1C1:R5C2	The values in the first five rows, the first two columns.
R3C2:R4C4	The values in rows 3 and 4, columns 2 through 4.
R1:R100	The values in the first 100 rows, every column.
C3:C4	The values in columns 3 and 4 for every row.

If you request more data than available memory can accommodate, Q+E returns the #NULL! error value. In this case, you can use multiple REQUEST functions to retrieve the data in segments. For example, the following Microsoft Excel macro code retrieves the first 100 rows of data, and then the second.

```
=REQUEST(chan,"R1:R100")
=REQUEST(chan,"R101:R200")
```

The maximum amount of data that can be transferred in a single operation is about 64K bytes.

Area Specification Options

You can append options to an area specification to specify other parameters.

To retrieve column headings and/or record numbers, use these options:

Use this option	To
HEADERS	Retrieve column headings as well as data.
ROWNUM	Retrieve the record number with each record.

Options follow the area specification, each preceded by a forward slash. For example, the following REQUEST function retrieves all records from the Query window, including column headings and row numbers:

```
=REQUEST(chan,"ALL/HEADERS/ROWNUM")
```

You can also add an option to change the file format used to transfer the data. Data is automatically transferred between Q+E and a client application using a default file format acceptable to both applications. If you add an option to override the default file format, first make sure that the client application supports the different file format. If you specify a format that is not supported by the client application, an error will result.

You can override the default file format using one of these options:

Use this option	To transfer data in this DDE format:
TEXT	Text format. This format transfers data using the tab character to separate values and a carriage return/linefeed combination to separate lines. Text format is supported by virtually all applications. Q+E supports Text format with all topic and item combinations.
BIFF	Binary file format (BIFF). Q+E data is automatically transferred to Microsoft Excel in this format. For more information, see the *Microsoft Excel User's Guide*.
RTF	Rich text format (RTF). This format transfers data in an RTF table. Q+E data is automatically transferred to Microsoft Word for Windows in this format. For more information, see the *Microsoft Word for Windows Technical Reference*.

NOTE   You can also use the HEADERS, ROWNUM, and file format options to modify the area specification item in a Fetch command. For more information, see "FETCH and FETCH.ADVISE" later in this chapter.

Retrieving Other Types of Information

In addition to area specifications, Q+E supports the following request items:

This item	Returns
NUMROWS	The number of records in the Query window.
NUMCOLS	The number of columns in the Query window.
SQLTEXT	The SQL SELECT statement for the Query window.
QUERY	The Query window name.
FIELDDEF	The expressions and data types of the columns in the Query window.
	For each column, the following information is returned: the column expression, its data type, the width of the column, the number of digits to the right of the decimal point (for numbers only), and the database-specific data type (varies depending on the source database system).

This item	Returns
	The data types returned will be one of the following:

Value	Data type
1	Character
2	Varying character
3	Decimal
4	Integer
5	Short integer
6	Short floating point
7	Floating point
8	Date

If the topic is SYSTEM, these request items are also available:

This item	Returns
SYSITEMS	The list of items supported by the SYSTEM topic.
TOPICS	The list of all currently open queries.
FORMATS	The list of all DDE formats supported.
STATUS	The current status of Q+E.

Sending Data

In addition to requesting data from Q+E with the REQUEST function, you can send data to Q+E using the POKE function. This function has three parameters:

Parameter	Description
channel_num	The value returned by the INITIATE function.
item	An area specification that identifies the field or range of fields in the Query window that you want to change.
data	The data sent to Q+E.

For example, the following Microsoft Excel macro statement changes the values in the first column of rows 1 and 2 in the Query window to the values stored in cells C1 and C2 in the macro sheet.

```
=POKE(chan,"R1C1:R2C1","C1:C2")
```

When you include a POKE statement in a Microsoft Word for Windows macro, the data parameter should contain tabs (ASCII character 9) to separate columns and carriage return/linefeed combinations (ASCII characters 13 and 10) to separate rows. For example, the following Microsoft Word for Windows macro statement sends four values to Q+E—two in the first row and two in the second.

```
DDEPoke chan,"Jones"+chr$(9)+"Bob"+chr$(13)+chr$(10)+"Smith"+chr$(9)+"Joe"
```

Executing Commands

You can send commands to Q+E using the EXECUTE function. This function has two parameters:

Parameter	Description
channel_num	The value returned by the INITIATE function.
execute_string	One or more commands to be executed by Q+E.

Each server application has a different set of "execute" commands that can be sent to it via DDE. Most of Q+E's execute commands correspond to the Q+E menu commands. The Q+E execute commands are described in detail in Chapter 11, "Q+E Execute Command Reference."

For example, the following Microsoft Excel macro statement tells Q+E to print the data in the Query window.

=EXECUTE(chan,"[PRINT(1,0)]")

The parameters "(1,0)" following the PRINT command tell Q+E to print one copy using regular (not draft quality) printing.

You can include more than one Q+E command in a single execute string. For example, this Microsoft Excel macro statement executes the Page Setup command and the Print command:

=EXECUTE(chan, "[PAGE.SETUP('Employees',1,1,2,2,1,0,1,0,0)][PRINT(1,0)]")

The parameters following the Page Setup command specify the report title, the page margins, and whether column headings and page numbers are to be included on each page. The Print command prints the data using the page setup.

Note that the entire execute string is a single parameter. Each command in an execute string must be enclosed in square brackets ([ ]).

If a parameter is a character value (such as 'Employees' in the example above), it must be enclosed in either single (') or double (") quotation marks. Many macro languages require double quotation marks around the entire execute string. In this case, you should use single quotation marks to enclose character values.

An EXECUTE statement returns an error if the channel number is invalid or if any errors occur when Q+E executes the commands in the execute string. The EXECUTE function does not normally return data to the client application.

FETCH and FETCH.ADVISE

All Q+E execute commands are described in detail in Chapter 11, "Q+E Execute Command Reference." However, two execute commands—FETCH and FETCH.ADVISE—deserve special mention. These commands retrieve data from Q+E to a worksheet or document.

The REQUEST function returns data to a macro program, which can then transfer the data to a worksheet or document. It is often more convenient, however, to

retrieve Q+E data using a Fetch command, since the Fetch command can transfer the data directly to a worksheet or document.

The following Microsoft Excel macro statement includes a FETCH command:

```
=EXECUTE(chan,"[FETCH('EXCEL','SHEET1','R1:R100','ALL')]")
```

To execute this statement, Q+E does the following:

□ Initiates a channel from Q+E to the worksheet named SHEET1.

□ Sends all the data in the Query window to row 1 through row 100 in SHEET1.

□ Terminates the channel.

FETCH.ADVISE is similar to FETCH, except the new channel is not terminated immediately. Q+E continues to send any changes to the data until the original DDE channel is closed or until Q+E receives a FETCH.UNADVISE command.

NOTE   Since Q+E must be able to send data to the client application, FETCH and FETCH.ADVISE can be used only from a client application that can also act as a server.

Closing a DDE Channel

The TERMINATE function closes a DDE channel. This function has only one parameter—the channel number of the DDE channel to be closed.

For example:

```
=TERMINATE(chan)
```

An error is returned if the channel number is invalid.

Examples

The following examples demonstrate how you can use DDE in macros and programs to exchange information with Q+E. All these examples are written using the Microsoft Excel macro language, although the same DDE functions and parameters would apply in other macro and programming languages.

Example 1

The following macro retrieves the first and last names of all employees in EMP.DBF:

```
=WORKSPACE(,,,,,,FALSE)
chan=INITIATE("QE","SYSTEM")
=EXECUTE(chan,"[OPEN('SELECT FIRST_NAME,LAST_NAME FROM EMP.DBF')]")
=EXECUTE(chan,"[FETCH('EXCEL','SHEET1','R3C1:R14C2','ALL')]")
=EXECUTE(chan,"[CLOSE()]")
=TERMINATE(chan)
=RETURN()
```

The WORKSPACE function ensures that the Ignore Remote Requests check box in the Microsoft Excel Options Workspace dialog box is turned off. It is a good idea to include this function at the beginning of any Microsoft Excel macro that includes a Fetch command. If Ignore Remote Requests is not turned off, Microsoft Excel will ignore any Fetch command that attempts to initiate a channel to it.

The INITIATE function creates a DDE channel between Microsoft Excel and Q+E. Since the topic is SYSTEM, the channel is not tied to a particular Query window.

The first EXECUTE function sends the OPEN execute command to Q+E. Q+E opens a Query window that contains the results of the SQL SELECT statement.

The next EXECUTE command sends a FETCH command to Q+E. This command transfers all the data in the Query window to rows 3 through 14, columns 1 and 2 in the Microsoft Excel worksheet named SHEET1.

The third EXECUTE function sends the CLOSE command to close the Query window.

NOTE   To run this macro more quickly, you could include OPEN, FETCH, and CLOSE in a single EXECUTE statement. An execute string in a Microsoft Excel macro can can be up to 255 characters long.

The TERMINATE function closes the DDE channel.

The RETURN statement ends the macro.

Example 2

In this macro, the channel to Q+E is initiated using the SYSTEM topic. This allows the macro to retrieve data from two different Query windows on one channel.

```
=WORKSPACE(,,,,,,,FALSE)
chan=INITIATE("QE","SYSTEM")
=EXECUTE(chan,"[OPEN('SELECT FIRST_NAME,LAST_NAME FROM EMP.DBF')]")
=EXECUTE(chan,"[FETCH('EXCEL','SHEET1','R1C1:R12C2','ALL')]")
=EXECUTE(chan,"[CLOSE()]")
=EXECUTE(chan,"[OPEN('SELECT DEPT_ID,DEPT_NAME FROM DEPT.DBF')]")
=EXECUTE(chan,"[FETCH('EXCEL','SHEET1','R20C1:R25C2','ALL')]")
=EXECUTE(chan,"[CLOSE()]")
=TERMINATE(chan)
=RETURN()
```

The employee names are placed in the first 12 rows of SHEET1 and the department names are placed in rows 20 through 25 of SHEET1.

It is easier to open several Query windows on a single channel than to initiate a separate channel for each.

Example 3

This macro opens DEPT.DBF and EMP.DBF, joins the two files, then retrieves the results into a Microsoft Excel worksheet.

```
=WORKSPACE(,,,,,,FALSE)
chan=INITIATE("QE","SYSTEM")
=EXECUTE(chan,"[OPEN('SELECT FIRST_NAME,LAST_NAME,DEPT FROM EMP.DBF')]")
=EXECUTE(chan,"[SELECT.AREA('C3')]")
=EXECUTE(chan,"[OPEN('SELECT DEPT_ID,DEPT_NAME FROM DEPT.DBF')]")
=EXECUTE(chan,"[SELECT.AREA('C1')][JOIN()]")
=EXECUTE(chan,"[FETCH('EXCEL','SHEET1','R1C1:R20C4','ALL')]")
=EXECUTE(chan,"[CLOSE()]")
=TERMINATE(chan)
=RETURN()
```

The first EXECUTE statement opens a Query window that displays the employee names and department numbers from EMP.DBF; the following SELECT.AREA command selects the DEPT column in EMP.DBF.

In the fourth EXECUTE statement, "[SELECT.AREA('C1',)][JOIN()]" selects the DEPT_ID column in DEPT.DBF and then joins EMP.DBF and DEPT.DBF based on the columns selected in the two windows.

The FETCH command retrieves the results of the join into SHEET1.

NOTE   To save time, you could also join EMP.DBF and DEPT.DBF in a single SELECT statement.

Example 4

The following macro returns the number of rows of data and number of columns of data in a Query window. This information is used to determine the size of the area on SHEET1 that the data will occupy.

```
chan=INITIATE("QE","SELECT * FROM EMP.DBF")
NR=REQUEST(chan,"NUMROWS")
NC=REQUEST(chan,"NUMCOLS")
=EXECUTE(chan,"[FETCH('EXCEL','SHEET1','R1C1:R"&NR&"C"&NC&"','ALL')]")
=TERMINATE(chan)
=RETURN()
```

NOTE   It is faster to first use the FETCH command to transfer the data into a large area on the worksheet and then use the REQUEST command to retrieve the number of rows (NUMROWS) and the number of columns (NUMCOLS). However, you would not know how large the data would be in advance.

Example 5

This macro shows how you can update and delete records using DDE. It first deletes the record for employee E10001 and then changes the salary for Rich Holcomb. It demonstrates two different methods for locating a record to change.

```
chan=INITIATE("QE","SELECT * FROM EMP.DBF WHERE EMP_ID='E10001'")
=EXECUTE(chan,"[ALLOW.EDIT(TRUE)][SELECT.AREA('C1')]")
=EXECUTE(chan,"[DELETE.RECORDS()]")
=TERMINATE(chan)
=RETURN()
chan=INITIATE("QE","SELECT LAST_NAME,SALARY FROM EMP.DBF")
=EXECUTE(chan,"[ALLOW.EDIT(TRUE)][SELECT.AREA('C1')][FIND('Holcomb')]")
=EXECUTE(chan,"[KEYS('{TAB}30000')]")
=TERMINATE(chan)
=RETURN()
```

In the first example, the SELECT statement retrieves only the record to be deleted. Then, the ALLOW.EDIT command permits changes to be made, the SELECT.AREA command selects the record, and the DELETE.RECORDS command deletes it.

In the second example, the SELECT statement retrieves all records from EMP.DBF. The SELECT.AREA command selects the LAST_NAME column. The FIND command searches the LAST_NAME column for the value "Holcomb" and selects it. The KEYS command moves the cursor to the SALARY column, then replaces Holcomb's current salary with "30000".

Using a SELECT statement to select only the records to be modified is generally the more efficient of the two methods. If you need to retrieve other records for display, or if different groups of records are likely to be modified, using the FIND command to find the records may be more efficient.

Chapter 11 Q+E Execute Command Reference

This chapter describes all Q+E execute commands. Most of these commands correspond to commands on the Q+E menus.

You can send execute commands to Q+E using the DDE EXECUTE function in a macro or programming language. Q+E executes the commands as if you chose them from a menu. For information on using the EXECUTE function, see Chapter 10, "Using Dynamic Data Exchange to Communicate with Q+E."

You can also add Q+E commands to the end of a query (.QEF) file. In this case, the commands are executed each time you open the query file. To add commands to a query file, open the query file in a text editor and enter the commands at the end of the file.

Syntax of Commands

Each command description begins with a main heading that shows the command's syntax. The command reference follows the rules for syntax and arguments used in the *Microsoft Excel Function Reference*. In headings, required arguments are *italic and bold* and optional arguments are *italic* but not bold. In text, all arguments are italic. Underline characters represent spaces between words; for example, *num_chars* is an argument name. For more information, see "Introduction" in the *Microsoft Excel Function Reference*.

Arguments can be:

▫ Numbers

▫ Text

▫ Logical values

A text argument must be enclosed in either single (') or double (") quotation marks.

The two logical values are TRUE or FALSE. If you prefer, you can use ON or 1 for TRUE, and OFF or 0 for FALSE.

Changing the Active Window

Most Q+E execute commands operate on the contents of the active window. When an execute command is sent, it should be appropriate for the active window. For example, if a Define window is active, it would not make sense to send the GOTO.RECORD command (equivalent to the Search Goto command in a Query window).

By default, the active window is the last one opened. You can use the ACTIVATE command to change the active window. If you are using DDE, you must open a channel using the SYSTEM topic if you want to open more than one Q+E window on that channel. When using the SYSTEM topic, you can only activate windows that were opened on that DDE channel.

The description of each execute command indicates the type of windows to which the command can be sent.

Selecting a Field, Column, or Row

Many of Q+E's execute commands operate on a selected field, column, or row. You can use the SELECT.AREA or SELECT.COLUMN command to select an area in a window.

Adding and Updating Records

You can add new records or update existing records using the execute commands.

The ADD.RECORD? command brings up a dialog box in which you can add new records. The PASTE.APPEND command adds new records using values from the Clipboard or values passed as an argument.

The FORM? command brings up a dialog box in which you can update records. You can also use the PASTE command to replace selected values with values from the Clipboard or values passed as an argument. In addition, the KEYS command can be used to update values.

Command Reference

The rest of this chapter describes each Q+E execute command in alphabetical order. Most of the examples show what you would enter in a query file to execute these commands.

ACTIVATE(windowname_text)

Activates a Query window or Define window.

Windowname_text is the name of the Query window or Define window to be activated.

Note

This command can only be used on DDE channels initiated with the SYSTEM topic.

Example

The following command sequence opens two Query windows, reactivates the first window, and then sorts by last name:

```
OPEN('EMP.DBF','dBASEFile')
OPEN('DEPT.DBF','dBASEFile')
ACTIVATE('Query1')
SELECT.COLUMN('LAST_NAME')
SORT.ASCENDING()
```

ADD.AFTER(field_values)

Similar to the Edit Add After command in a Define window. Adds a new field definition after a selected field. If no field is selected, adds a new field after the last field. For detailed information on the Edit Add After command, see online Help for Q+E.

Field_values contains values for the new field definition. It must be a single parameter with a tab character separating each pair of values (text format). If omitted, ADD.AFTER is equivalent to the Edit Add After command and opens a blank row for adding a new field definition.

Example

The following Microsoft Excel macro commands open EMP.DBF in a Define window and then add a new field named AGE that is numeric and three characters wide.

=EXECUTE (chan, "[DEFINE('EMP.DBF','dBASEFile')]")
=EXECUTE (chan, "[ADD.AFTER('AGE"&CHAR(9)&"NUMERIC"&CHAR(9)&"3"&CHAR(9)&"0')]")

ADD.BEFORE(field_values)

Similar to the Edit Add Before command in a Define window. Adds a new field definition before a selected field. If no field is selected, adds a new field before the first field. For detailed information on the Edit Add Before command, see online Help for Q+E.

Field_values contains values for the new field definition. It must be a single parameter with a tab character separating each pair of values (text format). If omitted, ADD.BEFORE is equivalent to the Edit Add Before command and opens a blank row for adding a new field definition.

ADD.CONDITION(logical_op,relational_op,value,case_sensitive)

Equivalent to the Select Add Condition command in a Query window. Adds a condition that records must meet to be selected. The condition applies to the currently selected column. For detailed information on the Select Add Condition command, see online Help for Q+E.

Logical_op is a number that specifies the type of connection when there is a previous selection condition. If *logical_op* is 1, the connection is AND; if *logical_op* is 2, the connection is OR.

Relational_op is a number from 1 to 8 that specifies the relational operator.

Relational_op	Relational operator
1	=
2	!=
3	<

Relational_op	Relational operator
4	<=
5	>
6	>=
7	LIKE
8	NOT LIKE

Value specifies the value used in the condition.

Case_sensitive is a logical value corresponding to the Case Sensitive check box in the Add Condition dialog box. If *case_sensitive* is TRUE, the selection condition is case sensitive; if FALSE, it is not.

Example

The following command sequence selects employees whose salary is less than $30,000:

```
OPEN('EMP.DBF','dBASEFile')
SELECT.COLUMN('SALARY')
ADD.CONDITION(1,3,30000,FALSE)
```

ADD.RECORD?()

Equivalent to the Edit Add Record command in a Query window. Displays a form for adding a new record. To use this command, editing must be enabled using the ALLOW.EDIT command. For detailed information on the Edit Add Record command, see online Help for Q+E.

Note

The following command can be added only to a query file; it is not available on a DDE channel.

ALLOW.EDIT(enable)

Equivalent to the Edit Allow Editing command in a Query window. Allows or disallows updating, adding, and deleting of records. For detailed information on the Edit Allow Editing command, see online Help for Q+E.

Enable is a logical value that specifies whether editing is enabled. If *enable* is TRUE, editing is allowed; if FALSE, it is not.

ARRANGE.ALL()

Equivalent to the Window Arrange All command in a Query window or Define window. Rearranges and resizes open windows so that you can see all of them at the same time. For detailed information on the Window Arrange All command, see online Help for Q+E.

CLOSE()

Equivalent to the File Close command in a Query window or Define window. Closes the active window. For detailed information on the File Close command, see online Help for Q+E.

Note

If you are executing this command via DDE and the DDE channel was opened with any Topic other than SYSTEM, this command also terminates the channel.

CLOSE.INDEX(index_file)

Equivalent to the File Close Index command in a Query window. Closes the dBASE index file named *index_file*. For detailed information on the File Close Index command, see online Help for Q+E.

Index_file is the name of the dBASE index file to be closed.

COLUMN.WIDTH(width,default_width)

Equivalent to the Layout Column Width command in a Query window. Changes the width of the selected columns. For detailed information on the Layout Column Width command, see online Help for Q+E.

Width is the number of characters or numeric digits that will fit in the column.

Default_width is a logical value specifying whether to make the selected columns the default width. If *default_width* is TRUE, the columns are set to the default width; if FALSE, the column widths are set to *width*.

Example

The following command sequence opens EMP.DBF in a Query window and then changes the width of FIRST_NAME to 10 characters:

```
OPEN('EMP.DBF','dBASEFile')
SELECT.COLUMN('FIRST_NAME')
COLUMN.WIDTH(10,FALSE)
```

COMMAND(op_num,exec_string)

When a Query window or Define window is active, sends a string of execute commands to Q+E.

Op_num is a number from 1 to 3 that specifies how Q+E receives the commands.

Op_num	Result
1	Q+E initializes a buffer and saves *exec_string*, but does not execute it.
2	The *exec_string* is concatenated to the buffer.
3	The *exec_string* is concatenated to the buffer, and the complete command buffer is executed.

Exec_string is the string of execute commands. Each command must be enclosed in square brackets ([ ]), and the entire string must be enclosed in single or double quotation marks. This is the same format that is used with the DDE EXECUTE function.

Notes

◻ This command cannot be included in a query file.

◻ This command exists because some macro languages, such as the Microsoft Excel macro language, have limits on the size of character variables. If you need to construct a Q+E execute command that is larger than the macro language's limit, it is convenient to send the command to Q+E in pieces.

COPY()

Equivalent to the Edit Copy command in a Query window or Define window. Copies selected text to the Clipboard. For detailed information on the Edit Copy command, see online Help for Q+E.

Example

The following command sequence copies the third record to the Clipboard:

```
SELECT.AREA('R3')
COPY()
```

COPY.SPECIAL(colhdr,rec_num,formula,data)

Equivalent to the Edit Copy Special command in a Query window or Define window. Copies data to the Clipboard according to the specified arguments. For detailed information on the Edit Copy Special command, see online Help for Q+E.

Colhdr is a logical value specifying whether to copy column headings. If *colhdr* is TRUE, column headings are copied; if FALSE, they are not.

Rec_num is a logical value specifying whether to copy record numbers. If *rec_num* is TRUE, record numbers are copied; if FALSE, they are not.

Formula is a number that specifies the type of external formula if the data is paste linked into another document. If *formula* is 1, the formula type is a query name. If *formula* is 2, the formula type is a SQL SELECT statement.

Data is a number from 1 to 3 that specifies what is copied to the Clipboard.

Data	Copied to Clipboard
1	All data
2	Selected data
3	SQL SELECT statement text

Example

The following command copies all the data in the active window to the Clipboard. If you used the Edit Paste Link command to paste the data into Microsoft Excel, the external formula would be the SQL SELECT statement describing the query. Column headings are included, but not record numbers.

COPY.SPECIAL(TRUE,FALSE,2,1)

CUT()

Equivalent to the Edit Cut command in a Query window or Define window. Moves the selected data to the Clipboard and deletes it in the window. For detailed information on the Edit Cut command, see online Help for Q+E.

DEFINE(name,source,charset_num)

Similar to the File Define command in a Query window or Define window. Opens a blank Define window for defining a new database file or displays the definition of an existing database file. For detailed information on the File Define command, see online Help for Q+E. To delete a file definition, use the DELETE command.

Name is the name of the database file you want to open in a Define window. If it is an empty string, Q+E opens a blank Define window for defining a new database file.

Source identifies the source database system. The values for *source* are the same as the source names in the File Define dialog box. If omitted, the current source is assumed.

Charset_num is a number that specifies which character set is used to store data in the file if the source is dBASEFile. If *charset_num* is 1, the ANSI character set is used. If *charset_num* is 2, the IBM PC character set is used. If omitted, the current default is assumed.

Examples

This command displays the definition of EMP.DBF. The file contents are stored using the IBM PC character set.

DEFINE('EMP.DBF','dBASEFile',2)

This command creates a blank Define window for creating a new dBASE database file.

DEFINE('','dBASEFile')

DEFINE.COLUMN(heading_text,expression_text,add,dest_num)

Similar to the Layout Define Column command in a Query window. Changes the definition of a selected column or adds a new column. For detailed information on the Layout Define Column command, see online Help for Q+E.

Heading_text specifies the column heading for the changed or new column. If blank, Q+E uses *expression_text* as the heading.

Expression_text is the expression that defines the values in the changed or new column.

Add is a logical value that specifies whether the selected column is changed or a new column is added.

- If *add* is TRUE, a new column is added in column *dest_num* with the heading *heading_text* and with values defined by *expression_text* .

- If *add* is FALSE, the selected column's heading is changed to *heading_text* and its expression is changed to *expression_text* .

Dest_num specifies the column number for the new column if *add* is TRUE. If *dest_num* is blank, the column is not added. This argument is ignored if *add* is FALSE.

Examples

The following command adds a LAST_NAME column as the second column, using the expression as the column heading:

DEFINE.COLUMN('','LAST_NAME',TRUE,2)

The following command sequence changes the expression for the LAST_NAME column so that it contains both first and last names, and changes the column heading to "Full Name":

SELECT.COLUMN('LAST_NAME')
DEFINE.COLUMN('Full Name','FIRST_NAME+LAST_NAME',FALSE)

DEFINE.FIELD(data_type,parm_1,parm_2)

Equivalent to the Layout Define Field command when a text file or Microsoft Excel worksheet file is displayed in a Query window. Changes the field specification of selected columns in a text file or worksheet file. For detailed information on the Layout Define Field command, see online Help for Q+E.

Data_type is a number that specifies the type of data in the column.

If the source is Textfile:

Data_type	Type of data
1	Character
2	Numeric
3	Date

If the source is ExcelFile:

Data_type	Type of data
1	Character
2	Floating point
3	Integer
4	Date
5	Logical

Parm_1 is a number that specifies the maximum width or format of the values in the selected columns.

If the source is TextFile:

◻ If *data_type* is 1 or 2 (character or numeric values), *parm_1* is the maximum number of characters or digits that will be displayed in the column.

◻ If *data_type* is 3 (date values), *parm_1* is the character mask describing the format of the dates. The mask can be any of those listed in the Layout Define Field dialog box.

If the source is ExcelFile:

◻ If *data_type* is 1 (character values), *parm_1* is the maximum number of characters that will be displayed in the column. Otherwise, this value is ignored.

Parm_2 is the number of digits to the right of the decimal point when *data_type* is 2 (numeric) and the source is TextFile. If the source is ExcelFile, this value is ignored.

Examples

The following command sequence specifies that the first column contains character values with a maximum length of 20 characters:

```
SELECT.AREA('C1')
DEFINE.FIELD(1,20)
```

The following command sequence specifies that the second and third columns contain numeric values up to 10 digits long with 2 digits to the right of the decimal point:

```
SELECT.AREA('C2:C3')
DEFINE.FIELD(2,10,2)
```

The following command sequence specifies that the fourth column contains date values formatted like 12/31/89:

```
SELECT.AREA('C4')
DEFINE.FIELD(3,'mm/dd/yy')
```

DEFINE.INDEX(index_file,tag,expression,unique,desc,op_num)

Equivalent to the File Define Index command in a Define window. Creates, rebuilds, or deletes a dBASE index file. For detailed information on the File Define Index command, see online Help for Q+E.

Index_file is the name of the index file that contains the index to create, rebuild, or delete.

Tag is the name of the index tag to create, rebuild, or delete if *index_file* is a dBASE IV (.MDX) file.

Expression is the index expression used to create or rebuild the index.

Unique is a logical value that specifies whether the index is created or rebuilt as a unique index. If *unique* is TRUE, the index will be unique; if FALSE, it will not.

Desc is a logical value that specifies whether the index is created or rebuilt in descending order. If *desc* is TRUE, the index will be in descending order; if FALSE, it will not.

Op_num is a number that specifies whether a new index is created or an existing index is rebuilt or whether an index is deleted. If *op_num* is 1, the index is created or rebuilt. If *op_num* is 2, the index is deleted.

Example

The following command creates or rebuilds a dBASE III index file on the LAST_NAME field. The index will not be unique, and the last name values will be in ascending order.

DEFINE.INDEX('EMPLNAME.NDX','','LAST_NAME',FALSE,FALSE,1)

DELETE(name,source)

Deletes a database file. To define or modify a database file definition, use the DEFINE command.

Name is the name of the database file to delete.

Source identifies the source database system. The values for *source* are the same as the source names in the File Define dialog box.

DELETE.FIELDS()

Equivalent to the Edit Delete Fields command in a Define window. Deletes the selected fields. For detailed information on the Edit Delete Fields command, see online Help for Q+E.

DELETE.RECORDS()

Equivalent to the Edit Delete Records command in a Query window. Deletes the selected records from the database. For detailed information on the Edit Delete Records command, see online Help for Q+E.

Example

The following command sequence deletes the record whose EMP_ID column contains E10001:

```
SELECT.COLUMN('EMP_ID')
FIND('E10001')
DELETE.RECORDS()
```

DISTINCT(enable)

Equivalent to the Select Distinct command in a Query window. When the source is SQL Server, Oracle, or Extended Edition Data Manager, hides duplicate rows in the Query window. For detailed information on the Select Distinct command, see online Help for Q+E.

Enable is a logical value specifying whether duplicate rows are hidden. If *enable* is TRUE, only unique rows appear; if FALSE, all rows are redisplayed.

ECHO(enable)

In a Query window or Define window, controls whether Q+E updates the screen after each command is executed.

Enable is a logical value that specifies whether the screen is updated after each execute command. If TRUE, Q+E redraws the screen after each command; if FALSE, the screen is not redrawn until:

- On a DDE channel, ECHO(ON) is executed or the DDE channel is terminated.
- After the last command from a query file is executed, ECHO is then re-enabled.

Until the ECHO command is used and *enable* is specified, TRUE is the default.

EXIT()

Equivalent to the File Exit command. From any window type, quits Q+E, automatically closing all open windows. For detailed information on the File Exit command, see online Help for Q+E.

FETCH*(dest_app,dest_topic,dest_item,qe_item)*

When a Query window is active, causes Q+E to send query results to a document or worksheet in another application. For an introduction to this command, see "Executing Commands" in Chapter 10, "Using Dynamic Data Exchange to Communicate with Q+E."

Dest_app is the application to which the data is to be sent (the destination).

Dest_topic is the worksheet or document to which the data is to be sent.

Dest_item is a reference to the area of the worksheet or document to which the data is to be sent.

Qe_item is a reference to an area that specifies what Q+E data is to be sent.

Notes

- FETCH cannot be used to send data to a Microsoft Word for Windows 1.0 document.

- FETCH cannot be included in query files.

- If the destination is Microsoft Excel, *dest_topic* must be the name of an open worksheet and *dest_item* must be an area specification in R*n*C*m* notation. For example, you would refer to the upper-left worksheet cell as R1C1, not as A1.

Example

The following command retrieves the first seven columns of the first six records into the area R10C1:R15C7 in the Microsoft Excel window SHEET1:

FETCH('EXCEL','SHEET1','R10C1:R15C7','R1C1:R6C7')

FETCH.ADVISE*(dest_app,dest_topic,dest_item,qe_item)*

This command is similar to FETCH, and its arguments are the same. However, if the data identified by *qe_item* changes before the next FETCH.UNADVISE or CLOSE command is executed, Q+E automatically resends the data. This ensures that the destination application always has the most up-to-date information. For information on the arguments to FETCH.ADVISE, see the description of the FETCH command above.

Example

The following command retrieves the first seven columns of the first six records into the area R10C1:R15C7 in the Microsoft Excel window SHEET1. Any later changes to the data in the Query window are automatically sent to SHEET1 until a FETCH.UNADVISE or CLOSE command is executed.

FETCH.ADVISE('EXCEL','SHEET1','R10C1:R15C7','R1C1:R6C7')

FETCH.UNADVISE(dest_app,dest_topic,dest_item)

In a Query window, cancels the previously executed FETCH.ADVISE with matching values for each of the three arguments.

Note

This command cannot be included in query files.

Example

The following command cancels the FETCH.ADVISE associated with the area R10C1:R15C7 in the Microsoft Excel window SHEET1:

FETCH.UNADVISE('EXCEL','SHEET1','R10C1:R15C7')

FIND(string_text)

Equivalent to the Search Find command in a Query window. Searches the selected columns for the first field containing *string_text* and selects that field. For detailed information on the Search Find command, see online Help for Q+E.

String_text is the string of characters to be found. The search is not case sensitive—uppercase and lowercase letters are treated identically.

Example

The following command searches for "Holcomb" in the LAST_NAME column:

SELECT.COLUMN('LAST_NAME')
FIND('Holcomb')

FIND.NEXT()

Equivalent to the Search Find Next command in a Query window. Finds the next occurrence of the search string specified by the most recent FIND command. The same columns are searched as in the previous FIND command. For detailed information on the Search Find Next command, see online Help for Q+E.

FIND.PREVIOUS()

Equivalent to the Search Find Previous command in a Query window. Finds the previous occurrence of the search string specified by the most recent FIND command. The same columns are searched as in the previous FIND command. For detailed information on the Search Find Previous command, see online Help for Q+E.

FONT(font_name,size_num,bold,italic,underline,strike,set_default)

Equivalent to the Layout Font command in a Query window. Changes the formatting of the text in the Query window. For detailed information on the Layout Font command, see online Help for Q+E.

Font_name is the font name.

Size_num is the point size.

The next five arguments are logical values corresponding to check boxes in the Layout Font dialog box. If an argument is TRUE, Q+E turns on the corresponding check box; if FALSE, Q+E turns off the check box.

Bold specifies whether characters are bold.

Italic specifies whether characters are italic.

Underline specifies whether characters are underlined.

Strike specifies whether characters are struck through.

Set_default specifies whether the new settings become the default for windows that are subsequently opened.

Example

The following command changes the text in the active Query window to the SYSTEM font, point size 8, with bold characters:

FONT('SYSTEM',8,TRUE,FALSE,FALSE,FALSE,FALSE)

FORM?()

Equivalent to the Edit Form command in a Query window. Opens a form-style dialog box that displays the data in the currently selected record. If no record is selected, the first record is displayed. For detailed information on the Edit Form command, see online Help for Q+E.

Note

This command is not available through DDE.

FORM.SETUP(max_col,max_flds_per_col,max_fld_width)

Equivalent to the Edit Form Setup command in a Query window. Defines a custom layout for the form-style dialog box displayed by subsequent FORM? and ADD.RECORD? commands. For detailed information on the Edit Form Setup command, see online Help for Q+E.

Max_col is the maximum number of columns in the dialog box.

Max_flds_per_col is the maximum number of fields in each column.

Max_fld_width is the maximum number of characters that will fit in each field's edit box.

Example

The following command sets the limits for the next form to have no more than 2 columns of fields, 10 fields per column, and a maximum edit box width of 20 characters:

FORM.SETUP(2,10,20)

GOTO.RECORD*(rec_num)*

Equivalent to the Search Goto command in a Query window. Selects the first field of the record whose record number is *rec_num*. For detailed information on the Search Goto command, see online Help for Q+E.

JOIN*()*

Equivalent to the Select Join command in a Query window. Joins the data in the top two Query windows using the common values selected in each window. For more information on joins, see Chapter 2, "Selecting and Sorting Data." For detailed information on the Select Join command, see online Help for Q+E.

Note

This command can be issued only on a channel initiated with the SYSTEM topic.

Example

The following command sequence joins EMP.DBF to DEPT.DBF using the common values in their Department columns:

```
OPEN('EMP.DBF','dBASEFile')
SELECT.COLUMN('DEPT')
OPEN('DEPT.DBF','dBASEFile')
SELECT.COLUMN('DEPT_ID')
JOIN()
```

KEYS*(keys)*

When a Query window or Define window is active, sends one or more keystrokes to Q+E to be executed.

Keys is a string of one or more keystrokes Q+E will execute. In addition to numbers and text, you can send the following special keys:

Special key	Meaning
{TAB}	TAB key
{LEFT}	LEFT ARROW
{RIGHT}	RIGHT ARROW
{UP}	UP ARROW
{DOWN}	DOWN ARROW

Special key names must be enclosed in braces ({ }).

Example

Assuming that a HIRE_DATE field in EMP.DBF is selected and editing has been allowed, this command moves the selection to the next column, SALARY, and changes the value to 30000.

```
KEYS('{TAB}30000')
```

LOGOFF(dbname)

Equivalent to the File Logoff command in a Query window or Define window. Logs off from the database system named *dbname*. For detailed information on the File Logoff command, see online Help for Q+E.

Dbname specifies the system to log off from and can be any database system name listed in the File Logoff dialog box.

Example

The following command logs off from SQL Server:

LOGOFF('SQLServer')

LOGON(dbname)

Equivalent to the File Logon command in a Query window or Define window. Displays the Logon dialog box for logging on to the database system named *dbname*. For detailed information on the File Logon command, see online Help for Q+E.

Dbname specifies the system to log on to and can be any database system name listed in the File Logon dialog box.

Example

The following command displays a dialog box for logging on to SQL Server:

LOGON('SQLServer')

MOVE.COLUMN(dest_num)

Similar to the Layout Move Column command in a Query window. Moves the selected columns to *dest_num*. For detailed information on the Layout Move Column command, see online Help for Q+E.

Dest_num is the position number of the leftmost selected column (1 is the first column, 2 the second, and so on). If more than one column is selected, the additional columns are positioned to the right of the *dest_num* column.

Example

The following command sequence selects the second column and moves it to the fourth column.

SELECT.AREA('C2')
MOVE.COLUMN(4)

ONLY.SHOW.TOTALS(enable)

Equivalent to the Layout Only Show Totals command in a Query window. If column totals are specified using the TOTALS command, this command can hide the records used to compute the totals. For detailed information on the Layout Only Show Totals command, see online Help for Q+E.

Enable is a logical value that specifies what is displayed. If *enable* is TRUE, only total values are displayed; if FALSE, both records and total values are displayed.

OPEN(spec_text,source,use_colhdr,charset_num, sql_num,set_default)
Form 1: When source is dBASEFile

OPEN(spec_text,source,use_colhdr,charset_num, file_type,delim_text,hdr_line,set_default, scan_lines,guess_type)
Form 2: When source is TextFile

OPEN(spec_text,source,use_colhdr,scan_lines,guess_type,set_default)
Form 3: When source is ExcelFile

OPEN(spec_text,source)
Form 4: When source is SQLServer, Oracle, or Extended Edition

Equivalent to the File Open command. When any window type is active, displays in a Query Window the contents of a database file or the results of a query file or SQL SELECT statement. For detailed information on the File Open command, see online Help for Q+E.

Spec_text can be the name of a dBASE database file, text file, Microsoft Excel worksheet file, query file, or a SQL SELECT statement that queries any source supported by Q+E. (*Spec_text* cannot be just a table name for SQL Server, Extended Edition, or Oracle).

Source specifies the source for *spec_text*. It can be any source listed in the File Open dialog box. If *spec_text* is a filename with an extension that identifies the source (for example, .DBF implies dBASE file and .QEF implies query file), the filename extension overrides *source*. If omitted, the current source is assumed.

Use_colhdr is a logical value specifying whether to use the column headings as field names in the database file. If *use_colhdr* is TRUE, the column headings are used as field names. If *use_colhdr* is FALSE, the underlying database field names are used. If omitted, the current default is assumed.

Charset_num is a number that determines which character set is used to store data in the database file. If *charset_num* is 1, the ANSI character set is used; if it is 2, the IBM PC character set is used. If omitted, the current default is assumed.

Sql_num is a number that determines which version of SQL is used. If *sql_num* is 1, ANSI SQL is used; if it is 2, dBASE IV SQL is used. If omitted, the current default is assumed.

File_type specifies whether the text file is character delimited or fixed format:

- If the file contains character-delimited values, *file_type* must be 1.

- If the file contains fixed-length values, *file_type* must be 2.

If omitted, the current default is assumed.

Delim_text specifies the delimiter character if *file_type* is 1. If the delimiter is the tab character, enter the value "TAB". This argument is ignored for fixed-format files. If omitted, the current default is assumed.

Hdr_line   is a logical value specifying whether to use the first line in the file as column headings.

❏ If *hdr_line* is TRUE, Q+E uses the values in the first line for column headings.

■ If *hdr_line* is FALSE, the column headings are FIELD_1, FIELD_2, and so on.

If omitted, the current default is assumed.

Set_default   is a logical value specifying whether the values set for all arguments except *spec_text* and *source* will be the defaults whenever you open dBASEFile, TextFile, or ExcelFile. If *set_default* is TRUE, the argument values you specify will be the defaults; if FALSE, they will not. If omitted, the current default is assumed.

Scan_lines   is a number specifying the number of records Q+E will scan to guess an appropriate width for each column in the Query window. If *source* is TextFile, *file_type* must have a value of 1 (character delimited). If *guess_type* is TRUE, Q+E also assigns a data type to each column based on the values in these records. If *scan_lines* is 0, Q+E scans all records in the file. If omitted, the default is 25.

Guess_type   is a logical value specifying whether Q+E will assign a data type to each column based on the values in the number of records specified by *scan_lines*. If *source* is TextFile, *file_type* must have a value of 1 (character delimited).

■ If *guess_type* is TRUE, Q+E guesses the data types.

■ If *guess_type* is FALSE, Q+E assigns the character data type to all columns.

If omitted, the current default is assumed.

Note

This command can be issued only on a channel initiated with the SYSTEM topic.

Examples

The following command opens a query file, executes it, and displays the query results in a new window:

OPEN('C:\WINDOWS\EMP.QEF')

The following command executes an SQL SELECT statement on EMP.DBF and displays the results in a Query window using ANSI SQL.

OPEN('SELECT * FROM dBASEFile|EMP',,,1)

or

OPEN('SELECT * FROM EMP','dBASEFile',,1)

OPEN.INDEX(index_file,use)

Equivalent to the File Open Index command in a Query window. Opens the dBASE index file named *index_file*. For detailed information on the File Open Index command, see online Help for Q+E.

Index_file is the name of the dBASE index file to open.

Use is a logical value specifying whether the index is to be used to sort the records if it is a dBASE III (.NDX) index file. If *use* is TRUE, the index is used to sort; if FALSE, the index is opened and maintained but not used to sort.

If the index file is a dBASE IV (.MDX) file, this argument is ignored.

Note

For information on using dBASE indexes to sort and improve performance, see Appendix A, "Using Q+E with dBASE."

Example

The following sequence of commands opens EMP.DBF, opens the dBASE III index built on the HIRE_DATE field, and uses the index to display the employee records ordered by hire date.

```
OPEN('EMP.DBF','dBASEFile')
OPEN.INDEX('EMPHIRE',TRUE)
```

OUTER.JOIN()

Equivalent to the Select Outer Join command in a Query window. Joins the data in the top two Query windows using the common values selected in each window. For more information on joins, see Chapter 2, "Selecting and Sorting Data." For detailed information on the Select Join command, see online Help for Q+E.

Note

This command can be issued only on a channel initiated with the SYSTEM topic.

PAGE.SETUP*(title_text,Lmargin_num,Rmargin_num,Tmargin_num,Bmargin_num,headings,rec_num,page_num,date,time)*

Equivalent to the File Page Setup command in a Query window. Sets the formatting options for subsequent PRINT and SAVE.TEXT.AS commands. For detailed information on the File Page Setup command, see online Help for Q+E.

Title_text is the title to be displayed centered on the first line of each page.

Lmargin_num specifies the left margin in inches.

Rmargin_num specifies the right margin in inches.

Tmargin_num specifies the top margin in inches.

Bmargin_num specifies the bottom margin in inches.

The next five arguments are logical values corresponding to check boxes in the File Page Setup dialog box. If an argument is TRUE, Q+E turns on the corresponding check box; if FALSE, Q+E turns off the corresponding check box.

Headings specifies whether column headings are included.

Rec_num specifies whether record numbers are included.

Page_num specifies whether page numbers are included at the top right of each page.

Date specifies whether the date is included at the top left of each page.

Time specifies whether the time is included at the top left of each page, underneath the date.

Example

The following command sequence sets up this page format: title is "Employee Information", the page has 1 inch top and bottom margins and .75 inch left and right margins, and column headings and page numbers are included.

PAGE.SETUP('Employee Information',.75,.75,1,1,TRUE,FALSE,TRUE,FALSE,FALSE)

PARSE.LINE(string_text)

Equivalent to the Layout Parse Line command in a Query window. Specifies how to parse a fixed-format text file so that values are displayed in separate fields. For detailed information on the Layout Parse Line command, see online Help for Q+E.

String_text contains square brackets ([ ]) showing the byte offset and length of each field in each line. All other characters in the string are ignored. The brackets themselves take up no space.

Example

The following command specifies a parse string, where the first five characters make up the first field and the next four characters make up the second field.

PARSE.LINE('[12345][6789]')

PASTE(field_values)

Similar to the Edit Paste command in a Query window or Define window. Pastes the values from the Clipboard or pastes *field_values* into the selected area. For detailed information on the Edit Paste command, see online Help for Q+E.

Field_values is optional and contains values for fields of the record. If omitted, the new values are pasted from the Clipboard. *Field_values* is a single argument with tabs separating field values (text format).

Examples

The following command sequence copies columns 3 through 5 of the second record into columns 3 through 5 of the third record:

SELECT.AREA('R2C3:R2C5')
COPY()
SELECT.AREA('R3C3:R3C5')
PASTE()

The following Microsoft Excel macro commands change the name in the first employee record to Teddy Andrews:

```
=EXECUTE(chan,"[OPEN('EMP.DBF','dBASEFile')]"
=EXECUTE(chan,"[ALLOW.EDIT(TRUE)]"
=EXECUTE(chan,"[SELECT.AREA('R1')]"
=EXECUTE(chan,"[PASTE('Teddy"&CHAR(9)&"Andrews')]"
```

In Microsoft Excel macros, you type CHAR(9) to enter a tab character.

PASTE.APPEND(field_values)

Similar to the Edit Paste Append command in a Query window or Define window. Adds new records or field definitions. For detailed information on the Edit Paste Append command, see online Help for Q+E.

Field_values is optional and contains values for the records or field definitions. If omitted, the new values are pasted from the Clipboard. *Field_values* is a single argument with tabs separating field values (character-delimited text format).

Examples

The following command sequence creates three new records that are copies of records 2 through 4:

```
SELECT.AREA('R2:R4')
COPY()
PASTE.APPEND()
```

The following Microsoft Excel macro commands open a Query window on EMP.DBF and add a new employee with the name Edward Johnson, employee id E98765.

```
=EXECUTE(chan,"[OPEN('EMP.DBF','dBASEFile')]"
=EXECUTE(chan,"[ALLOW.EDIT(TRUE)]"
=EXECUTE(chan,"[PASTE.APPEND('Edward"&CHAR(9)&"Johnson"&CHAR(9)&"E98765')]"
```

PRINT(copies_num,draft)

Equivalent to the File Print command in a Query window or Define window. Prints the contents of the window in the format specified using the PAGE.SETUP command. For information on the File Print command, see online Help for Q+E.

Copies_num is the number of copies to print.

Draft is a logical value specifying whether the data is printed in draft mode. If *draft* is TRUE, the data is printed using draft quality; if FALSE, the data is printed using high quality.

Example

The following command prints one copy of the window contents in high quality:

```
PRINT(1,FALSE)
```

QUERY.NOW()

Equivalent to the Select Query Now command in a Query window. Executes the current SQL SELECT statement and displays the results. This command is used for shared database systems to see recent changes. Also, if you modify fields or add or delete records, QUERY.NOW ensures that computed columns are up to date, the sort order is correct, and so on. For detailed information on the Select Query Now command, see online Help for Q+E.

REMOVE.COLUMN()

Equivalent to the Layout Remove Column command in a Query window. Removes the selected columns from the display. For detailed information on the Layout Remove Column command, see online Help for Q+E.

Example

The following command sequence removes the second and third columns:

```
SELECT.AREA('C2:C3')
REMOVE.COLUMN()
```

SAVE()

When a query file is open and active, saves the current query definition.

SAVE.AS(name,dest_text,use_colhdr,type,charset_num)
Form 1: For Query windows when the destination database file is dBASEFile

SAVE.AS(name,dest_text,use_colhdr,type,charset_num,delimter,hdr_line)
Form 2: For Query windows when the destination database file is TextFile

SAVE.AS(name,dest_text,use_colhdr)
Form 3: For Query windows when the destination is ExcelFile, SQLServer, Oracle, or Extended Edition

When a Query window is active, saves the query results to a database file.

Name is the name for the new database file.

Dest_text is the database system that the file is to be stored in. It can be any destination database system listed in the File Save As dialog box.

Use_colhdr is a logical value specifying whether to use the column headings as field names in the new database file.

- If *use_colhdr* is TRUE, the column headings are used as field names.
- If *use_colhdr* is FALSE, the underlying database field names are used.

If omitted, the current default is assumed.

Type is a number that determines the file format.

When *dest_text* is dBASEFile:

Type	dBASEFile format
1	dBASE II
2	dBASE III
3	dBASE IV

When *dest_text* is TextFile:

Type	TextFile format
1	Character delimited
2	Fixed format

Charset_num determines the character set that is used to store data in the database file. If *charset_num* is 1, the ANSI character set is used. If *charset_num* is 2, the IBM PC character set is used. If omitted, the current default is assumed.

Delimiter is the delimiter character if *type* is 1 (character delimited). If the delimiter is the tab character, enter the value "TAB". This parameter is ignored for fixed-format files. If omitted, the current default is assumed.

Hdr_line is a logical value specifying whether the first line of the text file will contain the field names. If *hdr_line* is TRUE, the first line contains field names; if FALSE, it does not. If omitted, the default is assumed.

Examples

The following command saves the records in the active Query window to EMPNEW.DBF, formatted as a dBASE III file (type 2), using the ANSI character set (*charset_num* 1):

SAVE.AS('EMPNEW.DBF','dBASEFile',FALSE,2,1)

The following command saves the records in the active Query window to EMPNEW.CSV, formatted as a character-delimited text file (format 1), using the IBM PC character set (*charset_num* 2). The comma is the delimiter character and the first line of the file will contain field names.

SAVE.AS('EMPNEW.CSV','TextFile',FALSE,1,2,',',TRUE)

The following command saves the records in the active Query window to EMPNEW.TXT, formatted as a fixed-format text file (format 2), using the ANSI character set (*charset_num* 1). Field names will not be written to the file.

SAVE.AS('EMPNEW.TXT','TextFile',FALSE,2,1,,FALSE)

SAVE.AS*(name,type,pack,charset_num)*
Form 4: For Define Windows

When a Define window is active, saves the current database file definition. Either modifies an existing database file definition or creates a new database file. Equivalent to the File Save As command in a Define window. For detailed information on the File Save As command, see online Help for Q+E.

Name is the name of the file being defined. If the file exists, it is modified. Otherwise, Q+E creates a new file. The destination database system is the same as the source database system specified for the Define window.

Type is a number from 1 to 3 that determines the file format for a dBASE file.

Type	dBASEFile format
1	dBASE II
2	dBASE III
3	dBASE IV

Pack is a logical value specifying whether a dBASE file is to be packed, removing deleted records and reclaiming unused space. If *pack* is TRUE or omitted, the file is packed; if FALSE, it is not.

Charset_num determines the character set that is used to store data in a dBASE file. If *charset_num* is 1, the ANSI character set is used. If *charset_num* is 2, the IBM PC character set is used.

SAVE.LABELS.AS*(name,def_text,lines_between,start_col1,start_col2,start_col3,start_col4, charset_num)*

When a Query window is active, saves the query results to a text file formatted as mailing labels. Equivalent to the File Save As command when the destination is MailingLabels. For detailed information on the File Save As command, see online Help for Q+E.

Name is a name for the text file that will contain the mailing labels. Q+E will add the extension .LAB automatically. If *name* is blank, the label definition given by *def_text* becomes the current definition, but no label file is generated. If *name* is the only argument, Q+E creates a label file using the current label definition.

Def_text describes the fields on each line of the label. Use the plus sign (+) to concatenate values; use a semicolon (;) to separate lines. Enclose text constants in single or double quotation marks.

Lines_between is the number of vertical lines between each label.

Start_col1, start_col2, start_col3, start_col4 are numbers that indicate the starting column positions for printing up to four columns of labels across the width of the page.

Charset_num determines the character set that is to be used to store data in the text label file. If *charset_num* is 1, the ANSI character set is used. If *charset_num* is 2, the IBM PC character set is used.

Example

The following command sequence creates a label file where the first line contains first name and last name, and the second line contains city, state, and zip with a comma between city and state:

SAVE.LABELS.AS ('ADDR.LAB','FIRST_NAME+LAST_NAME;CITY+","+STATE+ZIP',5,1,0,0,0)]

SAVE.QUERY.AS*(name)*

When a Query window is active, saves the current query definition to a query (.QEF) file. Equivalent to the File Save As command when the destination is QueryFile. For detailed information on the File Save As command, see online Help for Q+E.

Name is a name for the new query file. Q+E will add the extension .QEF automatically.

Example

The following command saves the current query to the file EMP.QEF:

SAVE.QUERY.AS('EMP.QEF')

SAVE.TEXT.AS*(name,width,lines,charset_num)*

When a Query window is active, saves the query results to a file formatted as text. Equivalent to the File Save As command when the destination is PrintToFile. For detailed information on the File Save As command, see online Help for Q+E.

Name is a name for the new text file. Q+E will add the extension .TXT automatically.

Width is the maximum number of characters that will fit on each line.

Lines is the maximum number of lines per page.

Charset_num determines the character set that is used to store data in the text file. If *charset_num* is 1, the ANSI character set is used. If *charset_num* is 2, the IBM PC character set is used.

Example

The following command saves the query results to EMP.TXT, formatted as text, with lines no longer than 80 characters, and no more than 60 lines per page:

SAVE.TEXT.AS('EMP.TXT',80,60)

SELECT.AREA(rowcol_text)

When a Query window or Define window is active, selects the specified rows and columns.

Rowcol_text   is an area specification in the format R*n*C*m*:R*x*C*y*. For more information on area specifications, see "Requesting Information" in Chapter 10, "Using Dynamic Data Exchange to Communicate with Q+E."

Example

The following command selects records 1 through 6, columns 1 through 4:

SELECT.AREA('R1C1:R6C4')

SELECT.COLUMN(col_expr_text)

When a Query window or Define window is active, selects a column based on its column expression.

Col_expr_text   is the column expression for the column that you want to select.

Example

The following command selects the column containing last name values:

SELECT.COLUMN('LAST_NAME')

SELECT.DELETED.RECORDS(enable)

Equivalent to the Select Select Deleted Records command in a Query window. Displays dBASE records that are marked for deletion or redisplays dBASE records that are not marked for deletion. For detailed information on the Select Select Deleted Records command, see online Help for Q+E.

Enable   is a logical value specifying whether records marked for deletion are displayed.

- If *enable* is TRUE, Q+E displays the dBASE records that are marked for deletion.
- If *enable* is FALSE, Q+E redisplays the dBASE records that are not marked for deletion.

SELECT.RESET()

Equivalent to the Select Reset Conditions command in a Query window. Removes all selection conditions from the current query and displays all records in the database file.

SORT.ASCENDING()

Equivalent to the Sort Ascending command in a Query window or Define window.

If a Query window is active, sorts the query results based on the selected columns. Records are sorted from smallest to largest.

If a dBASE file definition is open in a Define window, sorts the underlying database file by the selected fields. Sorting does not occur until the file definition is saved.

For detailed information on the Sort Ascending command, see online Help for Q+E.

Example

The following command sequence sorts the records in a Query window using the values in the fourth column:

```
SELECT.AREA('C4')
SORT.ASCENDING()
```

SORT.DESCENDING()

Equivalent to the Sort Ascending command in a Query window or Define window.

If a Query window is active, sorts the query results based on the selected columns. Records are sorted from largest to smallest.

If a dBASE file definition is open in a Define window, sorts the underlying database file by the selected fields. Sorting does not occur until the file definition is saved.

For detailed information on the Sort Ascending command, see online Help for Q+E.

Example

The following command sequence sorts the records in the LAST_NAME column:

```
SELECT.COLUMN('LAST_NAME')
SORT.DESCENDING()
```

SORT.RESET()

Equivalent to the Sort Reset Sorting command in a Query window or Define window.

If a Query window is active, removes all sorting conditions from the current query and displays the records in their original order in the database file.

If a dBASE file definition is open in a Define window, removes all sorting conditions from the database file. When the definition is saved, the order of the records cannot be reset.

For detailed information on the Select Reset Sorting command, see online Help for Q+E.

SQL.QUERY(stmt_text)

Equivalent to the Select SQL Query command in a Query window. Replaces the current SQL SELECT statement with a new one and displays the results in the Query window. For detailed information on the Select SQL Query command, see online Help for Q+E.

Stmt_text is the new SQL SELECT statement.

Example

The following command replaces the contents of the Query window with the records from the EMP.DBF database file:

SQL.QUERY('SELECT * FROM EMP.DBF')

TOTALS(min,max,count,avg,sum)

Equivalent to the Layout Totals command in a Query window. Adds or removes totals for the selected columns. For detailed information on the Layout Totals command, see online Help for Q+E.

The arguments are logical values corresponding to the check boxes in the Layout Totals dialog box.

- If an argument is TRUE, Q+E calculates and displays the total.
- If an argument is FALSE, Q+E removes the total.

Min displays the minimum value.

Max displays the maximum value.

Count counts the values and displays the total.

Avg calculates and displays the average value.

Sum calculates and displays the sum of the values.

Note

The average and sum arguments are valid only for numeric fields.

Example

The following command sequence displays the minimum and maximum value in the third column:

SELECT.AREA(C3)
TOTALS(TRUE,TRUE,FALSE,FALSE,FALSE)

UNDO*()*

Equivalent to the Edit Undo command in a Query window or Define window. Reverses the action of the last command that can be undone. For detailed information on the Edit Undo command, see online Help for Q+E.

Example

The following command sequence adds a LAST_NAME column as the fifth column, then removes it:

```
DEFINE.COLUMN(",'LAST_NAME',TRUE,5)
UNDO()
```

UPDATE.ALL*(newval_text)*

Equivalent to the Edit Update All command in a Query window. Changes the value of all selected fields to the same value. For detailed information on the Edit Update All command, see online Help for Q+E.

Newval_text is the value that is to replace all the selected values.

Example

The following command sequence changes every value in the DEPT column to D101:

```
SELECT.COLUMN('DEPT')
UPDATE.ALL('D101')
```

USE.INDEX*(index_file)*

Equivalent to the File Use Index command in a Query window. Sorts dBASE records in the Query window using a dBASE index. For detailed information on the File Use Index command, see online Help for Q+E.

Index_file is the name of the index file (no extension) or index tag that Q+E will use to sort the records in the Query window.

Note

For information on using dBASE indexes to sort and improve performance, see Appendix A, "Using Q+E with dBASE."

Example

The following command uses the index file EMPHIRE.NDX to sort the employee records by their hire date:

```
USE.INDEX('EMPHIRE.NDX')
```

Part 5

Appendixes

Appendix A   Using Q+E with dBASE

This appendix describes the Q+E features that are unique to dBASE-compatible files.

Q+E supports dBASE II-, dBASE III-, and dBASE IV-compatible database files. Each value in a dBASE database file is stored in a field, and fields are grouped to form records. All records in a database file must be of the same type. For example, the EMP.DBF database file contains employee records and DEPT.DBF contains department records. You cannot mix employee and department records in the same database file.

In addition to what is covered in the main part of this manual, this appendix provides information about:

- Opening and saving dBASE database files.
- Displaying records marked for deletion.
- Using dBASE index files to sort and improve performance.
- Creating, rebuilding, and deleting indexes.
- Defining dBASE database files.
- SQL supported by the Q+E dBASE driver.
- Expressions supported by the Q+E dBASE driver.
- Differences between ANSI SQL and dBASE IV SQL.

Opening and Saving dBASE Database Files

This section describes the options that are available when you open or save a dBASE file.

Options for Opening dBASE Files

When you open a dBASE database file in Q+E, you can choose the Options button in the File Open dialog box to specify these additional options:

- **File Character Set**   Select the character set you want Q+E to use when it stores data in the file. The two character sets are similar, although ANSI (American National Standards Institute) has better support for international characters. dBASE creates files using the IBM PC character set; Microsoft Excel creates files using the ANSI character set. If you are using Q+E exclusively to create and maintain database files, use the ANSI character set. The default is IBM PC.

- **SQL Compatibility**   Select the type of SQL you want Q+E to use to build the SQL SELECT statement for the Query window. The default SQL type is dBASE IV. For more information, see "Differences Between ANSI SQL and dBASE IV SQL" later in this chapter.

- **Set Default**   Turn on this check box if you want the current settings to be the default for dBASE files you open in the future.

Options for Saving dBASE Database Files

When you save a new dBASE file in Q+E, you can choose the Options button in the Save As dialog box to specify these additional options.

▫ **File Character Set**   Select the character set you want Q+E to use when it writes data to the file. For more information, see "Options for Opening dBASE Files" above. The default is IBM PC.

▫ **File Format**   Select the version of dBASE with which you want the file to be compatible. The default is dBASE IV.

Displaying Records Marked for Deletion

When you delete records from a dBASE file using the Edit Delete Records command, Q+E removes the records from the display, but they are not actually deleted. Instead they are "marked for deletion." You can switch between a display of the records marked for deletion and records not marked for deletion. If you remove a record and then change your mind, you can "undelete" it.

To permanently remove records marked for deletion, you must save the file definition in a Define window with the Pack File option selected. For more information, see "Defining dBASE Database Files" later in this appendix.

///// Switching between marked and unmarked records

▫ Choose Select Select Deleted Records.

When records marked for deletion are selected, a check mark appears next to the command on the menu. Choose Select Select Deleted Records again to display records not marked for deletion.

///// Restoring a record marked for deletion

1   Select any field in the record.

2   Choose Edit Undelete Records.

You can use the Select Add Condition command to select a subset of the records marked for deletion just as you can records that are not marked for deletion. Any selection conditions you apply to deleted records are applied to undeleted records when you return to the undeleted records, and vice versa.

Using dBASE Index Files

Q+E supports dBASE II, dBASE III, and dBASE IV index files. This section explains how to use a dBASE index to sort the records in a Query window or to improve performance when selecting records or joining files. Sample dBASE index files are included with Q+E so that you can try out the procedures in this section. For information on defining indexes, see "Creating, Rebuilding, and Deleting Indexes" later in this appendix.

About Indexes

An index is basically a two-column list that Q+E refers to internally. The first column refers to the records in a database file by key values in one or more columns. The second column contains the unique record numbers that point to the locations of the records. This is the same principle as an index in a book—you look up the subject (or value) you want, and the page number (or record number) tells you where to find it.

For example, you could use an index built on the LAST_NAME field of EMP.DBF to quickly sort the employee records by last name. Or, if you added a condition to the LAST_NAME column, such as LAST_NAME = 'Woltman', Q+E could look in this index to find the matching records more quickly.

You can have more than one index associated with a database file. You could have both a LAST_NAME index and an EMP_ID index for EMP.DBF—one to order the records by last name and the other to quickly locate records by employee number.

Indexes are stored in separate files. Each index is defined by an index expression whose values are used to build the index. The expression is usually a single field name, such as LAST_NAME in the example above. But an expression can be more complex. For example, if you sort using the index expression LAST_NAME + FIRST_NAME, Q+E sorts first on LAST_NAME and then on FIRST_NAME.

IMPORTANT   If you edit, add, or delete records in a database file, any associated indexes must also be updated. Otherwise, the indexes will not match the records in the database file. When you modify database files using Q+E, it automatically updates all open index files. If an index file is not open and you modify the database file, you need to rebuild the index before you can use it. For information on rebuilding indexes, see "Creating, Rebuilding, and Deleting Indexes" later in this appendix.

Using an Index to Join Records Faster

When you join database files, an index file can greatly improve performance.

For example, suppose you wanted to join EMP.DBF to DEPT.DBF based on their common Department column. To help Q+E join the files faster, you could open the EMPDEPT.NDX index file before you joined the files. EMPDEPT.NDX orders the employee records on the DEPT field. This allows Q+E to search the DEPT column in EMP.DBF and match values more quickly when it creates the join.

The records cannot be sorted using any index other than the index used to improve join performance. If the records are sorted on a different index, choose the File Use Index command and select [None] in the Use Index box.

Generally, you only open an index file for the source window of the join, not the destination window, and the index must be built on the field used to join the files. Having indexes open on other fields does not speed up the join.

Opening an Index File

Before you can use an index to sort or improve performance, the index file that contains the index must be open. Depending on your needs, you may want to have more than one index file open for the same database file.

dBASE II and III index files have the extension .NDX, and each file contains one index. dBASE IV index files have the extension .MDX, and each file may contain multiple indexes. Each index in a dBASE IV .MDX file is called a *tag* and has a tag name to identify it. By default, a dBASE IV index file has the same name as the associated database file and the .MDX extension.

IIIII   *Opening an index file*

When you open a dBASE IV database file in a Query window, Q+E automatically opens the .MDX index file with the same name as the database file. In all other cases, you must open the index files manually. EMP.DBF has associated with it three .NDX index files that must be opened manually.

1   Choose File Open and open the database file for which you want to open an index.

2   Choose File Open Index.

3   In the Open Index dialog box, type or select the name of the index file you want to open.

4   If you are opening a dBASE II or dBASE III (.NDX) index file, turn on the Use Index check box if you want to use the index to sort the records immediately. Otherwise, the index is opened and maintained, but not used for sorting. Turning on this check box has no effect when you open a dBASE IV (.MDX) index file.

5   Choose OK.

Using an Index to Sort

You can use a dBASE index file to display the records in the Query window in a predefined order. Using an index to sort records is faster than using the Sort commands but requires more disk space. For example, you could sort the records in EMP.DBF by hire date by using EMPHIRE.NDX or use EMPLNAME.NDX to sort the records by last name.

When you open a dBASE II or dBASE III index file, you can specify that you want to sort on the index at the same time you open it. Otherwise, you use the File Use Index command to select an index to sort by or to change the index used for sorting.

IIIII   *Selecting an index to sort by*

1   Choose File Use Index.

2   Select the index you want to use. If you do not want to use any index, select [None].

3   Choose OK.

Q+E reorders the records to correspond to the selected index.

Using an Index to Select Records Faster

When you select records using the Select menu commands or a SQL SELECT statement, an index may improve Q+E's performance.

For example, suppose you wanted to select the records in EMP.DBF with a DEPT value equal to D101. To help Q+E search for the records faster, you could open the index file named EMPDEPT.NDX before you added the selection condition. EMPDEPT.NDX orders the employee records on the DEPT field. This allows Q+E to search the DEPT column for "D101" more quickly.

The increase in speed may be negligible when you are working with a small number of records, as in EMP.DBF, but for larger database files, using an index can make the search time much shorter.

The ordering of items in a dBASE index is case sensitive. Therefore, turning on the Case Sensitive check box in the Add Condition dialog box will speed up the selection.

In addition, the records cannot be sorted using any index other than the index used to improve selection performance. If they are sorted on a different index, choose the File Use Index command and select [None] in the Use Index box.

Closing an Index File

Q+E closes all index files automatically when you close the database file.

IIIII   *Closing an index file manually*

1   Choose File Close Index.

2   Select the index file you want to close.

3   Choose OK.

Creating, Rebuilding, and Deleting Indexes

You can use Q+E to create new indexes, rebuild existing indexes, and delete old indexes. To create, rebuild, or delete a dBASE index, you first open the associated database file in a Define window and then use the File Define Index command.

If you are working with a dBASE IV .MDX file, you must specify the name of the .MDX file index tag that you want to create, rebuild, or delete.

Creating an Index

When you create a new index, you specify an index expression whose values are to be used to build the index. The expression is usually a single database field name, but it can include two or more field names concatenated by plus (+) signs.

Q+E also supports a number of dBASE functions that can be used in index expressions. For a list of the dBASE functions you can use, see "Expressions Supported by the Q+E dBASE Driver" later in this appendix.

▮▮▮▮ Creating a new index

1 Choose File Define and open the database file for which you want to create an index.

2 In the Define window, select the field on which you want to define an index. If you select two or more fields, select them in the order that you want them to appear in the index expression.

3 Choose File Define Index. Note that Q+E puts the selected field names in the Expression box. If you select more than one field, the fields are separated by plus (+) signs.

4 If necessary, edit the index expression.

5 If you want to create a dBASE II or dBASE III index, type a new index file- name with the extension .NDX in the File Name box.

   If you want to create a new dBASE IV index, you can create a new dBASE IV .MDX file to contain the index (or tag) or you can add the tag to an exist- ing .MDX file.

   To create a new .MDX file and tag, type a new index filename with the exten- sion .MDX in the File Name box and then type a tag name in the Tag Name box. If possible, give the index file the same name as the associated database file but use the .MDX extension. Then Q+E will automatically open and maintain the index file when you open the database file.

   To add the tag to an existing .MDX file, select the .MDX filename in the File List box and then type a tag name in the Tag Name box.

6 To create a unique index, turn on the Unique Index check box. A unique index does not include duplicate values. If you specify a unique index on a database file that contains records with duplicate index values, only the record with the first occurrence of the duplicate index value will be in the index. Unique indexes are not supported for dBASE II database files.

7 To create a dBASE IV index in descending order, turn on the Descending check box. By default, all indexes are created in ascending order (from small- est value to largest value).

8 Choose OK.

Now you can open and use the index when the associated database file is open in a Query window.

Rebuilding an Index

To ensure that an index is up to date with the associated database file, it must be rebuilt after you add or delete records from the database file and after you change the values of fields that are indexed. When you modify records in a Query window, Q+E automatically updates any index files that are open. Otherwise, you must rebuild an index yourself before you can use it.

///// Rebuilding an index

1   Choose File Define and open the database file for which you want to rebuild an index

2   Choose File Define Index.

3   In the File List box, select the index file you want to rebuild.

4   If you are rebuilding a dBASE IV (.MDX) index, select the tag you want to rebuild in the Tag List box. If you want to rebuild all the tags, clear the Tag Name and Expression boxes.

5   Choose OK.

You can also change the expression for an existing index in the Define Index dialog box and then rebuild it based on the new expression.

Deleting an Index

When you no longer need an index, it is a good idea to delete it to free up disk space.

///// Deleting an index

1   Choose File Define and open the database file for which you want to delete an index.

2   Choose File Define Index.

3   In the File List box, select the index file you want to delete.

4   If you are deleting a dBASE IV index, select the tag you want to delete in the Tag Name box. If you want to delete all the tags, clear the Tag Name and Expression boxes.

5   Choose the Delete button.

If you delete all the tags from a dBASE IV (.MDX) index file, the file is also deleted.

Defining dBASE Database Files

This section provides additional information about:

- Opening the definition of an existing dBASE database file.
- Defining fields for a dBASE database file.

□ Sorting the records in a dBASE database file.

□ Saving the definition of a dBASE database file, including packing to remove
   records marked for deletion and compressing any associated memo files.

Option for Opening dBASE File Definitions

When you open a dBASE file definition in a Define window, you can choose the
Options button in the File Define dialog box to specify the File Character Set
option as follows:

□ Select the character set you want Q+E to use when it writes data to the file.
   The default is IBM PC. For more information, see "Options for Opening
   dBASE Files" earlier in this appendix.

Defining Fields for a dBASE Database File

When you define a field for a dBASE database file in a Define window, Q+E
needs the following information:

Column	Definition
FIELD_NAME	The name of the field. Field names may contain up to 10 characters. You can use letters, numbers, or the underscore (_) character in names. dBASE II names may not contain underscores but may contain the colon (:) character. Names cannot contain blanks, and the first character of a field must be a letter.
TYPE	The field type can be:
	Character: contains letters, numbers, or any punctuation character found on your keyboard.
	Numeric: contains only numbers. This includes a decimal point and, optionally, a leading minus sign.
	Date: contains date values. Dates are not supported for dBASE II databases.
	Logical: used for true/false or yes/no information. The possible values are the letters T, F, Y, or N.
	Memo: used to store long, multiline textual data. This field type is not supported for dBASE II database files.
	Float: stored the same as numeric, but other programs may treat this field type differently. Float is valid for dBASE IV databases only.

Column	Definition
WIDTH	The number of characters required to hold the field's value.
	Character fields can be up to 254 characters long. Numeric and float fields can be up to 19 digits long. The width must include positions for all digits, a decimal point, and a leading sign if used.
	Date fields, logical fields, and memo fields are fixed in length. The widths for these fields are displayed automatically, and you cannot change them.
DECIMAL	The decimal value is used only for numeric and float fields. It indicates the number of digits to the right of the decimal point. It must be at least 2 less than the width in order to accommodate a leading 0 and a decimal point.

Adding a Memo Field

If you add a memo field to a dBASE database file (by selecting the "memo" data type), when you save the file, Q+E automatically creates a second file to hold any text entered into the memo column. The name of the memo file is the same as the associated database file, except the extension is .DBT instead of .DBF. When you modify a database file, Q+E automatically opens and maintains any memo files associated with the database file.

If you copy database files (.DBF) to another directory, be sure to copy the associated memo files (.DBT). If you rename a database file, rename the associated memo file.

Changing the Data Type of a Field in an Existing dBASE Database File

If you change a character field to a logical field in an existing dBASE database file, Q+E uses the leftmost character of the value in the character field to determine the value for the logical field. If the leftmost character is a Y, N, T, or F, then the logical field has that value. Otherwise, the logical field is blank.

If you change a logical, date, numeric, or float field to a character field, no information is lost unless the character field width is too small to contain the values.

Sorting Database Files

You can use the Define window to sort the records in a database file. This process is not the same as sorting records in a Query window. When you sort records in a Query window, the records are displayed in sorted order but the order of the records in the database file does not change. Sorting records in a Define window changes the order of the records in the database file.

You may want to sort a database file because you have other programs that read the database file and expect the records to be sorted. Or, you may want the database file sorted so the records are ordered the way you want them displayed when you open the file in a Query window.

To sort records in a Define window, use the Sort menu commands just as you do in a Query window. You can choose the Window Show Info command to view the current sort order.

Q+E does not actually sort the records in the database file until you save your changes using the File Save or File Save As command.

Options for Saving dBASE File Definitions

When you use the File Save As command to save a dBASE file definition, you can choose the Options button in the Save As dialog box to specify these additional options.

◻ **File Character Set** Select the character set you want Q+E to use when it writes data to the file. The default is IBM PC. For more information, see "Options for Opening dBASE Files" earlier in this appendix.

◻ **File Format** Select the version of dBASE with which you want the file to be compatible. The default is dBASE IV.

◻ **Pack File** Turn on this check box if you want Q+E to pack the database file and any associated memo files when it saves the database file. Packing removes records marked as deleted and compresses all the unused space from the files. If you delete many records without packing, you are wasting a lot of space in your file. In addition, if you edit memo fields a great deal, you may be wasting space. Memo text may be rewritten to the end of the memo file when it is changed. To use the file space occupied by the memo text before it was changed, you must pack the file.

SQL Supported by the Q+E dBASE Driver

The form of the SQL SELECT statement supported by the Q+E driver for dBASE files is:

```
SELECT <clause>
FROM <clause>
[ WHERE <clause> ]
[ ORDER BY <clause>]
[ COMPUTE <clause> ]
[ OPTIONS <clause> ]
```

The SELECT and FROM clauses are required. The other clauses (in brackets) are optional.

The syntax of each clause is described below. For information on using SQL to define Q+E queries, see Chapter 5, "Editing the Current SQL SELECT Statement."

SELECT Clause

The SELECT clause has the following form:

SELECT {* | column_expression, ...}

You can follow SELECT with an asterisk (*) to retrieve all fields or with a list of column expressions to retrieve only specified fields.

A column expression has the syntax *expression* [[=]*column_heading*]. *Expression* can be a field name (for example, LAST_NAME), or it can be an expression that defines a computed column.

Column_heading is the column heading that will appear in the Query window. If it is omitted, *expression* is used as the heading. If the heading contains blanks or other special characters, it must be enclosed in quotation marks.

If there is more than one file specification in the FROM clause, you can use *file*.* (for example, EMP.*) to retrieve all fields from one database file. Or, you can prefix a field name with the database filename (for example, EMP.LAST_NAME).

FROM Clause

The FROM clause has the form FROM *filespec*, ... Follow FROM with a list of file specifications to indicate from which files you want to select records.

A file specification has the syntax [dBASEFile|]*pathname* [*options*] [*table_alias*]. "dBASEFile|" indicates that the file being accessed is a dBASE-compatible file. If this prefix is omitted, dBASEFile must be the current source (selected in the File Open dialog box).

Pathname is the path and filename for the database file.

Options allows you to control the File Open options, whether deleted records are to be returned, and which index files are to be opened and used for sorting. The options specification is:

([COMPATIBILITY= {ANSI | DBASE},]
[CHARSET= {ANSI | IBMPC},]
[RECORDS= {DELETED | UNDELETED},]
[index_spec])

The COMPATIBILITY option determines whether ANSI- or dBASE IV-compatible SQL is to be used. The CHARSET option determines whether the database file uses the ANSI or IBM PC character set, and the RECORDS option determines whether undeleted or deleted records are to be returned. An example of a SQL SELECT statement using these options is:

SELECT * FROM EMP.DBF
(COMPATIBILITY=ANSI,CHARSET=IBMPC,RECORDS=UNDELETED)

Index_spec controls the use of index files. For dBASE II and III files, the form is *index_filename* [/USE], ... You list the .NDX files you want opened. If you add the /USE argument after a filename in the list, that index is also used to sort the records.

For dBASE IV files, *index_spec* looks like [*index_filename*] [*/tag_name*]. You list the .MDX files you want opened. Q+E will automatically open a .MDX file if it has the same name as the database file. The *tag_name* is the name of an index within the index file that you want to use to sort the records.

Here is an example using dBASE III index files:

SELECT * FROM EMP.DBF (EMPHIRE.NDX, EMPDEPT.NDX/USE)

Here is an example using a dBASE IV index:

SELECT * FROM EMP.DBF (/DEPTTAG)

Since no index filename is specified, it is assumed that EMP.MDX exists and contains an index named DEPTTAG.

Table_alias is a name you can use to refer to the file in the rest of the SELECT statement. You can prefix database field names with the table alias. Given the file specification:

FROM EMP.DBF E

you can refer to the LAST_NAME field as E.LAST_NAME. Table aliases must be used if the SELECT statement joins a table to itself. For example:

SELECT * FROM DEPT.DBF E, EMP.DBF F
WHERE E.MGR_ID = F.EMP_ID

WHERE Clause

To specify conditions that records must meet to be retrieved, add a WHERE clause with the form WHERE *conditions*.

Conditions are in the form *exp1 rel_operator exp2*. *Exp1* and *exp2* can be field names, constant values, or expressions. *Rel_operator* is the relational operator that links the two expressions.

For example, the following SELECT statement retrieves the names of employees whose salaries are at least $20,000:

SELECT LAST_NAME,FIRST_NAME FROM EMP.DBF WHERE SALARY >= 20000

ORDER BY Clause

To specify a sort order, add a ORDER BY clause with the syntax ORDER BY *sort_expression* [DESC | ASC], ... *Sort_expression* can be a field name, an expression, or the position of the column expression in the SELECT statement. For example, you could use either of these statements to sort EMP.DBF by LAST_NAME:

SELECT EMP_ID, LAST_NAME, FIRST_NAME FROM EMP.DBF ORDER BY LAST_NAME
SELECT EMP_ID, LAST_NAME, FIRST_NAME FROM EMP.DBF ORDER BY 2

In the second example, LAST_NAME is the second column expression following SELECT, so ORDER BY 2 sorts by LAST_NAME.

COMPUTE Clause

To compute column totals, add a COMPUTE clause with the form COMPUTE *total*,... . Q+E supports SUM, AVG, MIN, MAX, and COUNT. For example, this SELECT statement computes totals for the SALARY and HIRE_DATE columns in EMP.DBF:

SELECT SALARY,HIRE_DATE FROM EMP.DBF
COMPUTE SUM(SALARY),AVG(SALARY),MIN(HIRE_DATE)

OPTIONS Clause

If you want only the totals specified in the COMPUTE clause to be returned (not the records themselves), add this OPTIONS clause:

OPTIONS ONLY SHOW TOTALS

Expressions Supported by the Q+E dBASE Driver

Q+E's dBASE driver supports a varied set of operators and functions you can use in expressions. The values in expressions can be constants or can come from fields of the database records.

Constants

Constants are values which do not change and can be numbers, text, dates, or logical values. For example, in the expression PRICE * 1.05, the value 1.05 is a constant.

You must enclose text constants in pairs of single quotation marks (') or double quotation marks ("). To include a single quotation mark in a text constant enclosed by single quotation marks, use two single quotation marks together (for example, 'Don''t'). Similarly, if the constant is enclosed by double quotation marks, use two double quotation marks to include one.

You must enclose date constants in braces ({ }) (for example {01/30/89}). Date constants must be in the form MM/DD/YY.

The two logical constants are .T. for true and .F. for false.

Operators

dBASE supports different operators for numeric, character, and date expressions.

Operators in Numeric Expressions

You can include the following operators in numeric expressions:

Operator	Description
+	Addition
–	Subtraction
*	Multiplication
/	Division
**	Exponentiation
^	Exponentiation

You can precede numeric expressions with a unary plus (+) or minus (–).

For example, if SALARY is 20000, the expression –(SALARY * 1.1) returns the value –22000.

Operators in Character Expressions

You can include the following operators in character expressions:

Operator	Description
+	Concatenation keeping trailing blanks.
–	Concatenation moving trailing blanks to the end.

For example, if LAST_NAME is 'BENNETT ' and FIRST_NAME is 'TYLER ':

This expression	Returns this character string
FIRST_NAME + LAST_NAME	'TYLER BENNETT '
FIRST_NAME – LAST_NAME	'TYLERBENNETT '
FIRST_NAME – (' ' + LAST_NAME)	'TYLER BENNETT '

In the last example the two single quotation marks insert one space between the first and last names.

Operators in Date Expressions

You can include the following operators in date expressions:

Operator	Description
+	Adds a number of days to a date to produce a new date.
–	Shows the number of days between two dates, or subtracts a number of days from a date to produce a new date.

For example, if HIRE_DATE is {01/30/90}:

This expression	Returns this value
HIRE_DATE + 5	{02/04/90}
HIRE_DATE – {01/01/90}	29
HIRE_DATE – 10	{01/20/90}

Relational Operators

You can use one of the following relational operators to separate two expressions:

Operator	Description
=	Equal
<>	Not equal
!=	Not equal
#	Not equal
>	Greater than
>=	Greater than or equal
!<	Not less than (same as greater than or equal)
<	Less than
<=	Less than or equal
!>	Not greater than (same as less than or equal)
LIKE	Matching a pattern
NOT LIKE	Not matching a pattern
*=	Outer join; matches NULL for second value
=*	Outer join; matches NULL for first value
IS NULL	Equal to NULL
IS NOT NULL	Not equal to NULL
BETWEEN	Range of values between a lower and upper bound

For example, you could use relational operators to specify the following conditions in the WHERE clause of a SELECT statement:

```
SALARY <= 40000
DEPT = 'D101'
HIRE_DATE > {01/30/89}
SALARY + COMMISSION >= 50000
LAST_NAME LIKE 'Be%'
SALARY IS NULL
SALARY BETWEEN 10000 AND 20000
```

Logical Operators

You can use the following logical operators to combine two or more conditions:

Operator	Description
AND	Both conditions must be true.
OR	Either condition can be true.
NOT	Reverses the meaning of an expression.

For example, in the following expression both conditions must be true for a record to be selected:

SALARY = 40000 AND EXEMPT = .T.

The logical NOT operator is used to reverse the meaning. For example, in this case a record is selected if either or both conditions are false.

NOT (SALARY = 40000 AND EXEMPT = .T.)

Operator Precedence

The following table shows the order in which the operators are evaluated. The operators in the first line are evaluated first, then those in the second line, and so on. Operators on the same line are evaluated from left to right in the expression.

Precedence	Operators
1	Unary–, Unary +
2	**, ^
3	*, /
4	+, – (between numbers, dates, or character strings)
5	=, <>, !=, #, <, <=, !<, >, >=, !>, *=, =*, LIKE, NOT LIKE, IS NULL, IS NOT NULL, BETWEEN
6	NOT
7	AND
8	OR

The following example shows the importance of precedence:

WHERE SALARY > 40000 OR HIRE_DATE > {01/01/89} AND DEPT = 'D101'

Because AND is evaluated first, this query retrieves employees in department D101 hired after Jan 1, 1989, as well as all employees making more than $40,000, no matter what their department or hire date.

To force the clause to be evaluated in a different order, use parentheses to enclose the conditions that are to be evaluated first. For example, the following WHERE clause retrieves employees in department D101 that either make more than $40,000 or were hired after Jan 1, 1989:

WHERE (SALARY > 40000 OR HIRE_DATE > {01/01/89}) AND DEPT = 'D101'

Functions

Q+E supports a number of dBASE functions that you can use in expressions. In the following tables, the functions are grouped according to the type of result they return—character strings, numbers, or dates.

The following functions return character strings:

Use this function	To
RTRIM	Remove trailing blanks from a string. RTRIM('ABC ') returns 'ABC'.
TRIM	Remove trailing blanks from a string (same as RTRIM).
LTRIM	Remove leading blanks from a string. LTRIM(' ABC') returns 'ABC'.
UPPER	Make each letter of a string uppercase. UPPER('Rappl') returns 'RAPPL'.
LOWER	Make each letter of a string lowercase. LOWER('Rappl') returns 'rappl'.
LEFT	Return the specified number of leftmost characters of a string. LEFT('Woltman',3) returns 'Wol'.
RIGHT	Return the specified number of rightmost characters of a string. RIGHT('Woltman',4) returns 'tman'.
SUBSTR	Return a substring of a string. Parameters are the string, the first character to extract, and the number of characters to extract (optional). SUBSTR('Holcomb',2,3) returns 'olc', while SUBSTR('Holcomb',2) returns 'olcomb'.
SPACE	Return a string of blanks. SPACE(5) returns ' '.
DTOC	Convert a date to a character string. An optional second parameter determines the format of the result: 0 (the default) returns MM/DD/YY, 1 returns DD/MM/YY, 2 returns YY/MM/DD, 10 returns MM/DD/YYYY, 11 returns DD/MM/YYYY, and 12 returns YYYY/MM/DD. DTOC({01/30/89}) returns '01/30/89'. DTOC({01/30/89}, 0) returns '01/30/89'. DTOC({01/30/89}, 1) returns '30/01/89'. DTOC({01/30/89}, 2) returns '89/01/30'.
DTOS	Convert a date to a character string using the format YYYYMMDD. DTOS({01/23/90}) returns '19900123'.

Use this function	To
STR	Convert a number to a character string. Parameters are the number, the total number of output characters (including the decimal point), and the number of digits to the right of the decimal point (optional). STR(12.34567,4) returns '12'. STR(12.34567,4,1) returns '12.3' STR(12.34567,6,3) returns '12.346'.

The following functions return numbers:

Use this function	To
MOD	Divide two numbers and return the remainder of the division. MOD(10,3) returns 1.
LEN	Return the length of a string. LEN('ABC') returns 3.
MONTH	Return the month part of a date. MONTH({01/30/89}) returns 1.
DAY	Return the day part of a date. DAY({01/30/89}) returns 30.
YEAR	Return the year part of a date. YEAR({01/30/89}) returns 1989.

The following functions return dates:

Use this function	To
DATE	Return today's date. DATE() returns today's date.
CTOD	Convert a character string to a date. An optional second parameter specifies the format of the character string: 0 (the default) returns MM/DD/YY, 1 returns DD/MM/YY, and 2 returns YY/MM/DD. CTOD('01/30/89') returns {01/30/89}. CTOD('30/01/89',1) returns {01/30/89}.

Differences Between ANSI SQL and dBASE IV SQL

When you open a dBASE file in a Query window, you can choose the Options button in the File Open dialog box to specify which type of SQL you want Q+E to use to build the SQL SELECT statement for the Query window. By default, Q+E supports SQL as defined by dBASE IV, although you can also choose the standard ANSI SQL.

Generally, ANSI is the best choice. However, if you have learned SQL by using dBASE IV, you may want to use dBASE IV SQL since it does not always conform to the ANSI standard.

This section describes the differences between dBASE IV SQL and standard ANSI SQL.

Character Fields

If you compare a character field to a character constant in a WHERE clause (for example, LAST_NAME = 'S'), dBASE IV SQL returns every record whose last name begins with "S". ANSI SQL returns only those records with a last name that is exactly "S".

If you use the UPPER or LOWER function, dBASE IV SQL does not change the case of international characters (such as those with umlauts and accents).

NULL Values

A record's field has a NULL value if it has no value. ANSI SQL has special rules for NULL values, but dBASE IV SQL does not support them. If you choose ANSI compatibility, Q+E will treat blank field values as NULL.

The following chart shows how blank values are sorted depending on your choice of ANSI or dBASE IV SQL:

Data type	dBASE IV	ANSI
Number	Sorted as 0	Placed at front
Date	Sorted at end	Placed at front
Logical	Sorted as .F.	Placed at front
Character	Sorted as blanks	Placed at front

If a record has a blank field, ANSI and dBASE IV SQL differ in how the record is selected by a WHERE clause. For a field named X, the following conditions select records with blank values:

Data type	dBASE IV	ANSI
Number	X = 0	X is NULL
Date	X = { / / }	X is NULL
Logical	X = .F.	X is NULL
Character	X = ' '	X is NULL

Your choice of ANSI or dBASE IV SQL also influences the values of averages and counts. dBASE IV SQL averages blank numbers as if they were zero and counts all values. ANSI SQL does not average or count blank values.

The following chart gives an example of how dBASE IV and ANSI compute totals differently. In this example, five records have the following values in the AMOUNT field: 10, blank, 20, blank, 30. ANSI and dBASE IV SQL will compute the following totals for these values:

Total	dBASE IV	ANSI
SUM	60	60
MINIMUM	blank	10
MAXIMUM	30	30
AVERAGE	12	20
COUNT	5	3

Finally, dBASE IV and ANSI SQL generate different results when you have expressions that contain NULL values.

For numeric expressions involving a NULL value (for example, AMOUNT * 1.1 when AMOUNT is blank), dBASE treats blanks as zero. ANSI always returns the NULL value.

For date expressions involving a NULL value, ANSI always returns the NULL value. The following table shows how dBASE treats NULL values. In this example, HIRE_DATE is blank:

Expression	dBASE IV	ANSI
HIRE_DATE – {01/01/89}	0	NULL
HIRE_DATE + 10	blank	NULL
HIRE_DATE – 10	blank	NULL

For logical expressions, dBASE IV treats blank values as .F. and ANSI treats blank values as NULL.

The following chart gives the rules that ANSI applies when using AND and OR with NULL expressions.

Expression 1	Operator	Expression 2	End result
TRUE	AND	NULL	NULL
FALSE	AND	NULL	FALSE
TRUE	OR	NULL	TRUE
FALSE	OR	NULL	NULL
NULL	AND/OR	NULL	NULL

Appendix B   Using Q+E with Text Files

This appendix describes the Q+E features that are unique to text files.

Text files are readable ASCII files, which can be edited or printed using a text editor or word processor. There are two types of text files that you can open in a Query window:

Format	Description
Character delimited	Each line is a separate record. The field values on each line are separated by a special character (often a comma or a tab). Since a character separates values, each value can be a different length. Text files with comma-separated values often have the extension .CSV. When you open this type of file in Q+E, the delimiter character can be any printable character or the tab character.
Fixed format	Each line is a separate record. However, the field values on each line are not separated by a character. Instead, related values start at the same position on each line. If you display a fixed-format text file in a text editor, values appear in fixed-width columns.

You can use Q+E to open and query text database files just like regular database files. You cannot use Q+E to edit or define text files, although you can use the File Save As command to save query results as a new text file.

In addition to what is covered in the main part of this manual, this appendix provides information about:

□   Opening a text file.

□   Saving results to a new text file.

□   SQL supported by the Q+E Text File driver.

Opening a Text File

When you make "TextFile" the current source in the File Open dialog box, the File List box lists the files in the current directory with the extension .CSV. To see text files with a different extension, type *.*extension* in the File Name box and choose OK. For example, to view fixed-format text files in the current directory, you could type *.TXT in the File Name box and choose OK.

There are two sample text files included with Q+E—a character-delimited file named EMP.CSV and a fixed-format file named EMP.TXT.

After you select a text file in the File Open dialog box, you must choose the Options button to specify whether you are opening a character-delimited file or a fixed-format file. You can specify other options as well, such as using the values in the first line of the text file as column headings in the Query window.

If you open a character-delimited file, Q+E can scan the file and assign column widths and data types automatically.

If you open a fixed-format file, Q+E displays the data in one column and assigns it the character data type. You must use the Layout Parse Line command to parse the data so that it is displayed in separate columns, and you must use the Layout Define Field command to define the data type of each column.

Specifying Options for Opening a Text File

When you open a text file in Q+E, you must choose the Options button in the File Open dialog box to specify the following:

- **File Character Set** Select the character set you want Q+E to use to store the data in the file. The two character sets are similar, although ANSI (American National Standards Institute) has better support for international characters. Microsoft Excel creates files using the ANSI character set. If you are using Q+E exclusively to create and maintain database files, use the ANSI character set.

- **File Format** Select Character Delimited Values if you are opening a text file with character-separated values. Select Fixed-Length Values if you are opening a text file with fixed-length values.

- **Character Delimiter** Type the character delimiter in this box if you selected Character Delimited Values under File Length Values. For example, if values are separated by commas, enter a comma (,). If values are separated by the tab character, type TAB.

- **First Line Contains Field Names** Turn on this check box if you want the first line in the text file to be the column headings in the Query window. Field names are often located in the first line of a text database file.

- **Guess Data Type** If you are opening a character-delimited file, turn on this check box if you want Q+E to assign a data type to each column based on the values in the number of records indicated in the Number Records to Scan box. If you do not turn on this check box, Q+E assumes that all columns contain character values. After you open the text file, you can define data types directly using the Layout Define Field command. For more information, see "Changing Column Data Types in a Text File" later in this appendix.

- **Number of Records to Scan** If you are opening a character-delimited file, enter the number of records you want Q+E to scan to determine the data type for each column in the Query window. The default is 25. If you set this number to 0, Q+E will scan all the records in the text file—this guarantees that Q+E guesses the correct data types, but it may take a long time to scan the file.

- **Set Default** Turn on this check box if you want the current settings to be the defaults for any text files you open in the future.

For example, if you were opening EMP.CSV, you would select Character Delimited Values under File Format, enter a comma in the Character Delimiter box, turn on the First Line Contains Field Names check box, and turn on the Guess Data Type check box. When you open the file, the data appears in the Query window as it does for any other database file.

If you were opening EMP.TXT, you would select Fixed-Length Values under File Format and turn on the First Line Contains Field Names check box. When you open the file, all the data appears in the first column in the Query window. You must parse the data to display it properly and then assign appropriate data types.

Parsing the Data in a Fixed-Format File

When you open a character-delimited text file, Q+E automatically breaks each line of the file into separate fields according to the positions of the separator character. But when you open a fixed-format text file, such as EMP.TXT, there is no character to indicate where each field begins and ends. Therefore, Q+E treats each line in a fixed-format file as a value of one field in the first column. After you open a fixed-format file, you must use the Layout Parse command to break the data into separate fields.

▥▥▥   Parsing the data in a fixed-format file

To parse a fixed-format file, you parse only the first line in the file. Q+E automatically breaks the remainder of the records in the same way.

1   Choose Layout Parse Line. Q+E displays the first line from the file in the Parse Line box.

2   Mark where each value begins and ends by entering a [in front of each value and a] at the end of each value. Bracket pairs ([ ]) tell Q+E the position and length of each column in the Query window. For example, the correct parse line for EMP.TXT is:

[FIRST_NA][LAST_NAME][EMP_ID][HIRE_DATE][SALARY][DEPT][E]

If you want Q+E to guess where each value begins and ends, choose the Guess button. If you want to remove the brackets and start over, choose Clear.

3   Choose OK.

Q+E creates a column for each enclosed value.

Changing Column Data Types in a Text File

When you open a character-delimited text file, you can tell Q+E to guess the data types of the columns and to assign them automatically. For more information, see "Specifying Options for Opening a Text File" earlier in this appendix. Otherwise, Q+E assumes that all columns contain character values. You can use the Layout Define Field command to change the data types of the columns directly.

It is important to have the correct data type assigned to each column. Otherwise, some Q+E commands may not work properly. For example, suppose the SALARY column has the character data type. If you select the SALARY column and choose the Layout Totals command, the Totals dialog box will not contain the Average and Sum check boxes; Q+E assumes SALARY contains character values, not numbers, unless you specify the data type for SALARY as numeric.

ⅠⅠⅠⅠⅠ   *Changing column data types in a text file*

1   Select a field in each column for which you want to define the same data type.

2   Choose Layout Define Field.

3   Under Data Type, select the data type.

4   If you selected the Character or Number data type, enter in the Width box the maximum number of characters or digits you want to display in the column.

5   If you selected the Number data type, enter the number of digits to the right of the decimal point in the Decimal Digits box.

6   If you selected the Date data type, select the format you want in the Date Mask box. For more information, see "Selecting a Date Mask" below.

7   Choose OK.

Selecting a Date Mask

The characters in the Date Mask box represent the following:

Code	Description
dd	Day of the month.
mm	Month of the year using numbers. For example, 01 means January.
mmm	Month of the year using three-letter abbreviations. For example, JAN means January.
yy	Last two digits of the year, assumed to be in the twentieth century. For example, 90 means 1990.
yyyy	All four digits of the year.

Masks containing slashes (/) indicate that a character separates each part of the date. The character separator can be a slash (/), dash (-), period (.), or other character.

Masks that contain a slash will accept values without leading zeros. If there is no slash, leading zeros are required. The following table shows examples of acceptable values for different masks:

Mask	Acceptable values
mm/dd/yy	01/31/90 (January 31, 1990)
	2/1/89 (February 1, 1989)
	01.5.90 (January 5, 1990)
	2-15-90 (January 15, 1990)
mm/dd/yyyy	6/12/2001 (June 12, 2001)
mmddyy	013190 (January 31, 1990)
	13190 is not an acceptable value.
dd/mm/yy	15.4.90 (April 15, 1990)
ddmmmyy	04JUL90 (July 4, 1990)

Saving Results to a New Text File

When you save query results to a new text file, choose the Options button in the Save As dialog box to specify these options:

- **File Character Set**   Select the character set you want Q+E to use when it writes data to the file. The default is IBM PC.

- **File Format**   Select the type of values you want the file to contain. If you select Character Delimited Values, Q+E will separate values using the character you enter in the Character Delimiter box. If you select Fixed-Length Values, Q+E will use the data types of the columns in the Query window to determine the lengths.

- **Character Delimiter**   Type the character delimiter you want Q+E to use if you selected Character Delimited Values under File Format.

- **First Line Contains Field Names**   Turn on this check box if you want the first line of the output file to contain the field names.

Expressions and SQL Supported by the Q+E Text File Driver

Q+E's Text File driver supports the same expressions as the Q+E dBASE driver. For information on the operators and functions you can use to query text files, see "Expressions Supported by the Q+E dBASE Driver" in Appendix A, "Using Q+E with dBASE."

The Text File driver also supports the same SQL as the dBASE driver except for differences in the FROM clause:

- Use the prefix "TextFile!" in place of "dBASEFile!" to tell Q+E that the file you want to access is a text file.

- Use the *options* specification that is specific to text files (see below).

Otherwise, see "SQL Supported by the Q+E dBASE Driver" in Appendix A, "Using dBASE with Q+E," for a complete description of the SQL clauses you can use to query text files.

Options Specification for Text Files

When you use SQL to query a text file, the *options* specification in the FROM clause specifies how to open the file, parse the data in a fixed-format file, and define column data types.

The *options* specification for character-delimited files is:

```
([HEADERLINE= {0 | 1}, ]
[CHARSET= {ANSI | IBMPC}, ]
[DELIMITER= {TAB | 'character'}, ]
[data_type_spec])
```

The *options* specification for fixed-format files is:

```
([HEADERLINE= {0 | 1}, ]
[CHARSET= {ANSI | IBMPC}, ]
[PARSE= parse_string, ]
[data_type_spec])
```

The HEADERLINE option specifies whether the first line in the file contains field names that will be the column headings in the Query window. The CHARSET option determines whether the database file uses the ANSI or IBM PC character set. Each of these is optional. If CHARSET is not specified, the default is IBM PC. If HEADERLINE is not specified, the default is 0. An example of an SQL SELECT statement using these options is:

```
SELECT * FROM EMP.CSV (HEADERLINE=1,CHARSET=IBMPC)
```

For character-delimited files, the DELIMITER option determines the character used to separate values. Type TAB for tab-separated values; enclose any other character in quotation marks.

For fixed-format files, the PARSE option gives the parse string to be used to determine the position and length of each value in the file. The string should be identical to the string entered in the Layout Parse Line dialog box. For more information, see "Parsing the Data in a Fixed-Format File" earlier in this appendix.

Data_type_spec specifies the data type of each field in the file. The specification is:

```
{[CHAR(width) | NUMBER(width,[decimal_digits]) | DATE(mask)],...}
```

The parameters for each data type are the same as those given in the Layout Define Field dialog box. For more information, see "Defining Column Data Types in a Text File" earlier in this appendix.

For example, you could use the following SELECT statement to retrieve records from EMP.TXT:

```
SELECT LAST_NAME,HIRE_DATE,SALARY FROM TextFile|EMP.TXT (HEADERLINE=1,
CHARSET=IBMPC,PARSE="[FIRST_NA][LAST_NAME ][EMP_ID][HIRE_DATE ][SALARY ]
[DEPTE]",CHAR(20),DATE("MM/DD/YY"),NUMBER(10,2))
```

Appendix C   Using Q+E with SQL Server

This appendix describes the Q+E features that are unique to SQL Server.

Q+E supports the SQL Server database system sold by Sybase and Microsoft. The SQL Server database system uses a client/server architecture. You run Q+E on one or more client computers that are connected to a network. The database system runs on a separate server computer on the network.

In SQL Server, each value in a database table is stored in a field (or column). Fields are grouped to form records (or rows). A table contains a set of records of the same type (like a dBASE database file). Tables are placed in databases. You can have multiple databases on one server computer.

In addition to what is covered in the main part of this manual, this appendix provides information about:

▢ Logging on to SQL Server.

▢ Logging off from SQL Server.

▢ Opening SQL Server tables.

▢ Saving query results to a new SQL Server table.

▢ Editing records.

▢ Defining SQL Server tables.

▢ Expressions and SQL supported by the Q+E SQL Server driver.

Logging On

Before you can use Q+E to access SQL Server tables, you must log on to a database server. If you make SQL Server the default database system when you install Q+E, it automatically requests log on information the first time you try to open or define a database table. If SQL Server is not your default database system, use the File Logon command to access SQL Server.

///// Logging on to SQL Server

1 Choose File Logon.

2 Select SQLServer and choose OK.

3 In the Server name box, enter the name of the server computer that contains the tables you want to access.

4 In the Login ID box, enter your Login identification number.

5 In the Password box, enter your password.

6 Choose OK.

Logging Off

When you exit the Q+E program, Q+E automatically logs off from SQL Server. You can also log off from SQL Server manually if you want to free up resources while you are using Q+E.

IIIII *Logging off from SQL Server*

1 Choose File Logoff.

2 Choose SQLServer and choose OK.

Opening SQL Server Tables

When you make SQLServer the current source in the File Open dialog box, the dialog box displays options that are specific to SQL Server.

"Owner" is the current database name followed by the current user name. To change the owner, select an item in the Owner List box and choose OK. Database names are enclosed in square brackets ([ ]); the other names are user names.

The Table List box lists the tables, views, and stored procedures created by the current owner. Stored procedure names are preceded by an ampersand (@).

If the current user name is "dbo", the system tables are displayed in the Table List box. System tables begin with "sys".

Options for Opening SQL Server Tables

When SQL Server is the current source, you can choose the Options button in the File Open dialog box to specify which object types are displayed in the Table List box.

◻ **Listbox Entries** Turn on the check boxes for the objects you want displayed in the Table List box.

◼ **Set Default** Turn on this check box to make your choices the default for any SQL Server tables you open in the future.

Executing a Stored Procedure

A stored procedure consists of one or more Transact-SQL statements saved in a SQL Server table. To execute a stored procedure from Q+E, you open it like any other database table. The ampersand (@) is part of the stored procedure name.

If you open a stored procedure that contains a SELECT statement, Q+E executes the SELECT statement and displays the resulting records in a Query window. If there is more than one SELECT statement in the procedure, Q+E only displays the results of the first SELECT statement.

If you open a stored procedure that contains INSERT, UPDATE, or DELETE statements, Q+E adds, changes, or removes records from the specified tables.

Some stored procedures require arguments. To pass arguments to a stored procedure, enter their values in the Table Name box after the stored procedure name.

Saving Results to a New SQL Server Table

When you use File Save As to save query results to a SQL Server table, the current owner specifies the database name and user name for the new table.

☐ To create the table under a different owner, enter the name for the table in the form *owner_name.table_name*.

☐ To create the table in a different database, enter the name for the table in the form *database_name.owner_name.table_name*.

Editing Records

To edit, add, or delete records in a SQL Server table, you must have modify table privileges.

Defining SQL Server Tables

You can use the File Define command to create a new SQL Server table or to modify an exisiting database definition and save it as a new table. After you use Q+E to modify an existing table definition, you cannot save it back to the same table.

You can also use File Define to delete a SQL Server table if you have delete table privileges.

Defining Fields for a SQL Server Table

When you define a field for a SQL Server table in a Define window, Q+E needs the following information:

Column	Definition
FIELD_NAME	The name of the field. Field names may contain up to 30 characters. You can use letters, numbers, the underscore (_) character, the pound sign (#), or the dollar sign ($) in names, but the first character must not be the dollar sign.
TYPE	The field type can be any user-defined data types you have created, or one of the system-supplied data types:
	Char: contains letters, numbers, or any punctuation character found on your keyboard, up to 255 characters. It is fixed length.
	Varchar: contains letters, numbers, or any punctuation character found on your keyboard, up to 255 characters. It is variable length.

Column	Definition
	Text: contains long, multiline textual data, up to 2 billion characters. It is variable length. Q+E cannot display or edit text values of more than 10,000 characters.
	Int: contains integers between –2,147,483,648 and 2,147,483,647.
	Smallint: contains integers between –32,768 and 32,767.
	Tinyint: contains integers between 0 and 255.
	Bit: contains integers between 0 and 1.
	Float: contains floating-point values between 1.7E–308 to 1.7E+308 with 15 digits of precision.
	Money: contains dollar and cent values between ±922,337,203,685,447.5807 with accuracy to a ten-thousandth of a dollar.
	Datetime: contains date and time values. The earliest date that can be stored is January 1, 1753, and the last date that can be stored is December 31, 9999. The time accuracy is to .003 seconds.
	Timestamp: contains a timestamp automatically updated by SQL Server.
	Binary: contains up to 255 bytes of binary data. It is fixed length.
	Varbinary: contains up to 255 bytes of binary data. It is variable length.
	Image: contains binary data of up to 2 billion bytes. It is variable length. Q+E cannot display or edit image values of more than 10,000 characters.
WIDTH	The maximum number of characters or bytes allowed for this field's value.
	You specify width for char, varchar, binary, and varbinary data types. The other data types are fixed in length or do not require a width when they are declared.
REQUIRED	Specifies whether a value is required for this field. If not required, a field may be left blank.

Expressions and SQL Supported by the Q+E SQL Server Driver

You can use any expressions or SQL supported by SQL Server to define computed columns and to edit or write SQL statements in the SQL Query dialog box. For more information, see the documentation for SQL Server.

NOTE   If you prefer, you can also use the SQL that Q+E supports for querying dBASE files, text files, and Microsoft Excel worksheet files. For more information, see Appendix A, "Using Q+E with dBASE." In addition, you can use the DISTINCT keyword in the SELECT clause and add GROUP BY and HAVING clauses after the WHERE clause. When a Q+E SQL statement is sent to SQL Server, it is automatically translated into the SQL supported by SQL Server.

You can also prefix the first table name in the FROM clause of a SQL SELECT statement with "SQLServerl" to indicate that the SELECT statement should be sent to SQL Server. If this prefix is omitted, SQLServer must be the current source (selected in the File Open dialog box).

In addition, you can include a USE statement to identify the database to be used when processing the statement. For example, to display all authors from the database named "pubs", you would enter this statement:

USE pubs; SELECT * FROM authors

Appendix D   Using Q+E with Oracle

This appendix describes the Q+E features that are unique to Oracle.

The Oracle database system uses a client/server architecture. You run Q+E on one or more client computers that are attached to a network. The database system runs on a separate server computer on the network.

In Oracle, each value in a database table is stored in a field (or column). Fields are grouped to form records (or rows). A table contains a set of records of the same type (like a dBASE database file). Tables are placed in databases. You can have multiple databases on one server computer.

In addition to what is covered in the main part of this manual, this appendix provides information about:

▫ Logging on to Oracle.

▫ Logging off from Oracle.

▫ Opening Oracle tables.

▫ Saving query results to a new Oracle table.

▫ Editing records.

▫ Defining Oracle tables.

▫ Expressions and SQL supported by the Q+E Oracle driver.

Logging On

Before you can use Q+E to access Oracle tables, you must log on to a database server. If you make Oracle the default database system when you install Q+E, Q+E automatically requests log on information the first time you try to open or define a table. If Oracle is not your default database system, use the File Logon command to access Oracle.

▦ Logging on to Oracle

1   Choose File Logon.

2   Select Oracle and choose OK.

3   In the Server name box, enter the name of the server computer that contains the tables you want to access.

4   In the User Name box, enter your user name.

5   In the Password box, enter your password.

6   Choose OK.

Logging Off

When you exit the Q+E program, Q+E automatically logs off from Oracle. You can also log off from Oracle manually if you want to liberate memory resources while you are using Q+E.

ⅡⅡⅡ   Logging off from Oracle

1 Choose File Logoff.

2 Choose Oracle and choose OK.

Opening Oracle Tables

When you make Oracle the current source in the File Open dialog box, the dialog box contains options that are specific to Oracle.

"User Name" is the current user name. To change the user name, select an item in the User Name List box and choose OK.

The Table List box lists the tables, views, and synonyms created by the current user. Synonyms are alias names for tables. If the current user name is "dbo", the system tables are displayed in the Table List box. System tables begin with "sys".

Options for Opening Oracle Tables

When Oracle is the current source, you can choose the Options button in the File Open dialog box to specify which object types are displayed in the Table List box.

- **Listbox Entries**   Turn on the applicable check boxes for the objects you want displayed in the Table List box. The Procedures check box is dimmed.

- **Set Default**   Turn on this check box to make your choices the default for any Oracle tables you open in the future.

Saving Results to a New Oracle Table

When you use File Save As to save query results to an Oracle table, the current User Name specifies the user name for the new table.

To create the table under a different user, enter the name for the table in the form *user_name.table_name*.

Editing Records

To edit, add, or delete records in an Oracle table, you must have modify privileges.

Defining Oracle Tables

You can use the File Define command to create a new Oracle table or to modify an exisiting database definition and save it as a new table. After you use Q+E to modify an existing table definition, you cannot save it back to the same table.

You can also use File Define to delete an Oracle table if you have delete table privileges.

Defining Fields for an Oracle Table

When you define a field for an Oracle table in a Define window, Q+E needs the following information:

Column	Definition
FIELD_NAME	The name of the field. Field names may contain up to 30 characters. You can use letters, numbers, the underscore (_) character, the pound sign (#), or the dollar sign ($) in names, but the first character must be a letter.
TYPE	The field type must be one of the following data types:
	Char: contains letters, numbers, or any punctuation character found on your keyboard, up to 240 characters. It is variable length.
	Long: contains long, multiline textual data, up to 65,535 characters. It is variable length. Q+E cannot display or edit text values of more than 10,000 characters. You cannot add conditions to or sort on a field that has the long data type. Only one long field can appear in a single table.
	Number: contains numeric values in one of two forms:
	If WIDTH and DECIMAL values are not specified, contains floating point values with 40 digits of precision.
	If WIDTH and DECIMAL values are specified, DECIMAL indicates the number of digits to the right of the decimal point and WIDTH defines the maximum number of digits in the field.
	Date: contains date and time values. The earliest date that can be stored is January 1, 4712 B.C., and the last date that can be stored is December 31, 4712 A.D.
	Raw: contains up to 240 bytes of binary data. It is variable length.

Column	Definition
	Long Raw: contains up to 65,535 bytes of binary data. It is variable length. Q+E cannot display or edit long raw values of more than 10,000 characters. You cannot add conditions to or sort on a field that has the long raw data type. Only one long raw field can appear in a single table.
WIDTH	The maximum number of characters or bytes allowed for this field's value.
	You specify width for char, number, and raw data types. The other data types are fixed in length or do not require a width when they are declared.
DECIMAL	The number of decimal digits to the right of the decimal point. You only specify this for the number data type.
REQUIRED	Specifies whether a value is required for this field. If not required, a field may be left blank.

Expressions and SQL Supported by the Q+E Oracle Driver

You can use any expressions or SQL supported by Oracle to define computed columns and to edit or write SQL statements in the SQL Query dialog box. For more information, see the documentation for Oracle.

NOTE   If you prefer, you can also use the SQL that Q+E supports for querying dBASE IV files, text files, and Microsoft Excel worksheet files. For more information, see "SQL Supported by the Q+E dBASE Driver" in Appendix A, "Using Q+E with dBASE." In addition, you can use the DISTINCT keyword in the SELECT clause and add GROUP BY and HAVING clauses after the WHERE clause. When a Q+E SQL statement is sent to Oracle, it is automatically translated into the SQL supported by Oracle.

You can also prefix the first table name in the FROM clause of a SQL SELECT statement with "Oraclel" to indicate that the SELECT statement should be sent to Oracle. If this prefix is omitted, Oracle must be the current source (selected in the File Open dialog box).

Appendix E   Using Q+E with OS/2 Extended Edition

This appendix describes the Q+E features that are unique to IBM OS/2 Extended Edition database manager.

Extended Edition can be used on a standalone system or on a server computer attached to a network. When you run Q+E, you can access databases on your system as well as those on other computers that are attached to a network.

In Extended Edition, each value in a database table is stored in a field (or column). Fields are grouped to form records (or rows). A table contains a set of records of the same type (like a dBASE database file). Tables are placed in databases. You can have multiple databases on one server computer.

In addition to what is covered in the main part of this manual, this appendix provides information about:

□ Granting privileges.

□ Logging on to Extended Edition.

□ Logging off from Extended Edition.

□ Opening Extended Edition tables.

□ Saving query results to a new Extended Edition table.

□ Editing records.

□ Defining Extended Edition tables.

■ Expressions and SQL supported by the Q+E Extended Edition driver.

Granting Privileges

Each person who uses Extended Edition is assigned an authorization ID. There are three categories of authorization IDs: administrator, local administrator, and user. Members of the first two categories automatically have the privileges necessary to run Q+E. However, persons who have been assigned the "user" category must be granted EXECUTE and BIND privileges before they can use Q+E to access Extended Edition tables.

To grant user privileges, an administrator must run Query Manager or Q+E and execute the following SQL statement:

GRANT EXECUTE,BIND ON PROGRAM QE TO *authorization_list*

where *authorization_list* is the list of authorization IDs, separated by commas, or the single word PUBLIC to grant access to all users.

For more information, see the documentation for OS/2 Extended Edition.

Logging On

Before you can use Q+E to access Extended Edition tables, you must log on to a database. If you make Extended Edition the default database system when you install Q+E, it automatically requests log on information the first time you try to open or define a table. If Extended Edition is not your default database system, use the File Logon command to access Extended Edition.

IIIII **Logging on to Extended Edition**

1   Choose File Logon.

2   Select EEDataMgr and choose OK.

3   In the Database name box, enter the name of the database that contains the tables you want to access.

4   In the Authorization ID box, enter your user name.

5   In the Password box, enter your password.

6   Choose OK.

You can only log on to one database at a time. If you want to access tables in a different database, you must first log off and then log on to the new database.

Logging Off

When you exit the Q+E program, Q+E automatically logs off from Extended Edition. You can also log off from Extended Edition manually. If you are logged on to a database but want to access tables in a different database, you must log off from Extended Edition before you can log on to the other database.

IIIII **Logging off from Extended Edition**

1   Choose File Logoff.

2   Choose EEDataMgr and choose OK.

Opening Extended Edition Tables

When you make EEDataMgr the current source in the File Open dialog box, the dialog box contains options that are specific to Extended Edition database manager.

"Authorization ID" is the current user name. To change the current user name, select an item in the Authorization ID List box and choose OK.

The Table List box lists the tables and views created by the current user. If the current user name is "SYSIBM", the system tables are displayed in the Table List box. System tables begin with "SYS".

Options for Opening Extended Edition Tables

When EEDataMgr is the current source, you can choose the Options button in the File Open dialog box to specify which object types are displayed in the Table List box.

- **Listbox Entries**   Turn on the applicable check boxes for the objects you want displayed in the Table List box. The Procedures and Synonyms check boxes are dimmed.

- **Set Default**   Turn on this check box to make your choices the default for any Extended Edition tables you open in the future.

Saving Results to a New Extended Edition Table

When you use File Save As to save query results to an Extended Edition table, the current Authorization ID determines the user name for the new table.

To create the table under a different user name, enter the name for the table in the form *user_name.table_name*.

Editing Records

To edit, add, or delete records in an Extended Edition table, you must have modify privileges.

Defining Extended Edition Tables

You can use the File Define command to create a new Extended Edition table or to modify an existing database definition and save it as a new table. After you use Q+E to modify an existing table definition, you cannot save it back to the same table.

You can also use File Define to delete an Extended Edition table if you have delete table privileges.

Defining Fields for an Extended Edition Table

When you define a field for an Extended Edition table in a Define window, Q+E needs the following information:

Column	Definition
FIELD_NAME	The name of the field. Field names may contain up to 18 characters. You can use letters, numbers, or the underscore (_) character in names, but the first character must be a letter.
TYPE	The field type can be any one of the Extended Edition types:

Column	Definition
	Char: contains letters, numbers, or any punctuation character found on your keyboard, up to 254 characters. It is fixed length.
	Varchar: contains letters, numbers, or any punctuation character found on your keyboard, up to 4000 characters. It is variable length.
	Long Varchar: contains long, multiline textual data, up to 32,700 characters. It is variable length. Q+E cannot display or edit text values of more than 10,000 characters.
	Integer: contains integers between –2,147,483,648 and 2,147,483,647.
	Smallint: contains integers between –32,768 and 32,767.
	Float: contains double-precision floating-point values.
	Decimal: contains decimal numbers.
	Date: contains date values. This is a three-part value containing month, day, and year.
	Time: contains time values. This is a three-part value containing hour, minute, and second according to a 24-hour clock.
	Timestamp: contains date and time values. This is a seven-part value containing month, day, year, hour, minute, second, and microseconds.
WIDTH	The maximum number of characters or bytes allowed for this field's value.
	You specify width for char, varchar, and decimal data types. The other data types are fixed in length or do not require a width when they are declared.
DECIMAL	The number of decimal digits to the right of the decimal point. You only specify this for decimal data types.
REQUIRED	Specifies whether a value is required for this field. If not required, a field may be left blank.

Expressions and SQL Supported by the Q+E Extended Edition Driver

You can use any expressions or SQL supported by Extended Edition to define computed columns and to edit or write SQL statements in the SQL Query dialog box. For more information, see the documentation for OS/2 Extended Edition database manager.

NOTE   If you prefer, you can also use the SQL that Q+E supports for querying dBASE files, text files, and Microsoft Excel worksheet files. For more information, see "SQL Supported by the Q+E dBASE Driver" in Appendix A, "Using Q+E with dBASE." In addition, you can use the DISTINCT keyword in the SELECT clause and add GROUP BY and HAVING clauses after the WHERE clause. When a Q+E SQL statement is sent to Extended Edition, it is automa-tically translated into the SQL supported by Extended Edition.

You can also prefix the first table name in the FROM clause of a SQL SELECT statement with "EEDataMgrl" to indicate that the SELECT statement should be sent to Extended Edition. If this prefix is omitted, EEDataMgr must be the current source (selected in the File Open dialog box).

Appendix F   Using Q+E with Microsoft Excel Worksheet Files

This appendix describes the Q+E features that are unique to Microsoft Excel worksheet files.

Q+E's Microsoft Excel worksheet driver allows you to open a Microsoft Excel database in a Query window and query it like any other database file. You cannot use Q+E to edit or define Microsoft Excel worksheet files, although you can use the File Save As command to save query results as a new worksheet file.

In addition to what is covered in the main part of this manual, this appendix provides information about:

- Opening a Microsoft Excel worksheet in a Query window.

- Saving results to a new Microsoft Excel worksheet file.

- Expressions and SQL supported by the Q+E Microsoft Excel worksheet driver.

Opening a Microsoft Excel Worksheet in a Query window

To open a Microsoft Excel worksheet file in a Query window, the worksheet must be closed and it must contain a database range named "Database". Q+E automatically uses the values in the first row of the database range as column headings in the Query window.

If the worksheet was saved with a protection password, Q+E will prompt you for the password when you open it.

Options for Opening a Microsoft Excel Worksheet File

When you open a Microsoft Excel worksheet file in Q+E, you can choose the Options button in the File Open dialog box to specify these additional options:

- **Guess Data Type**   Turn on this check box if you want Q+E to assign a data type to each column based on the values in the number of records indicated in the Number of Records to Scan box. If you do not turn on this check box, Q+E assumes that all columns contain character values. After you open the worksheet file, you can define data types directly using the Layout Define Field command. For more information, see "Changing Column Data Types in a Microsoft Excel Worksheet File" later in this appendix.

- **Number of Records to Scan**   Enter the number of records you want Q+E to scan to determine the data type for each column in the Query window. The default is 25. If you set this number to 0, Q+E will scan all the records in the worksheet file—this guarantees that Q+E guesses the correct data types, but it may take a long time to scan the file.

- **Set Default**   Turn on this check box to make your choices the default for any Microsoft Excel worksheet files you open in the future.

Changing Column Data Types in a Microsoft Excel Worksheet File

When you open a Microsoft Excel worksheet file, you can tell Q+E to guess the data types of the columns and to assign them automatically (see above). Otherwise, it assumes that all columns contain character values. You can use the Layout Define Field command to change the data types of the columns directly.

It is important to have the correct data type assigned to each column. Otherwise, some Q+E commands may not work properly. For example, suppose the SALARY column has the character data type. If you select the SALARY column and choose the Layout Totals command, the Totals dialog box will not contain the Average and Sum check boxes; Q+E assumes SALARY contains character values, not numbers, unless you specify the data type for SALARY as float or integer.

Changing data types in a Microsoft Excel worksheet file

1 Select a field in each column for which you want to define the same data type.

2 Choose Layout Define Field.

3 Under Data Type, select the data type.

4 If you selected the Character data type, enter the maximum number of characters you want to display in the column in the Width box.

5 Choose OK.

Expressions and SQL Supported by the Q+E Microsoft Excel Worksheet Driver

Q+E's Microsoft Excel worksheet driver supports the same expressions as the Q+E dBASE driver. For information on the operators and functions you can use to query Microsoft Excel worksheet files, see "Expressions Supported by the Q+E dBASE Driver" in Appendix A, "Using Q+E with dBASE."

The Microsoft Excel worksheet driver also supports the same SQL as the dBASE driver except for differences in the FROM clause:

■ Use the prefix "ExcelFile!" in place of "dBASEFile!" to tell Q+E that the file you want to access is a Microsoft Excel worksheet file.

■ Use the options specification that is specific to Microsoft Excel worksheet files (see below).

Otherwise, see "SQL Supported by the Q+E dBASE Driver" in Appendix A, "Using dBASE with Q+E," for a complete description of the SQL you can use to query Microsoft Excel worksheet files.

Options Specification for Microsoft Excel Worksheet Files

When you use SQL to query a Microsoft Excel worksheet file, the options specification in the FROM clause specifies the data type of each field in the file.

The options specification for worksheet files is:

{[CHAR(*width*) | FLOAT | DATE | INTEGER | LOGICAL},...}

Width is the maximum number of characters you want to display in a character column and corresponds to the Width box in the Layout Define Field dialog box.

Index

A

Microsoft®
Making it all make sense™